



TUGAS AKHIR - TM 141585

RANCANG BANGUN SISTEM KONTROL PELONTAR PELURU DENGAN DUA SUMBU PUTAR

Irvan Dwi Putra
NRP. 2109 100 117

Dosen Pembimbing
Arif Wahjudi ST., MT., Ph.D.

JURUSAN TEKNIK MESIN
Fakultas Teknik Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015



TUGAS AKHIR - TM 141585

ARCHITECTURE BULLET LAUNCHER CONTROL SYSTEM WITH TWO-AXIS TURNTABLE

IRVAN DWI PUTRA
NRP. 2109 100 117

Academic Adviser
Arif Wahyudi ST., MT., Ph.D.

DEPARTMENT OF MECHANICAL ENGINEERING
FAKULTY OF INDUSTRIAL OF ENERGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA 2015

LEMBAR PENGESAHAN

RANCANG BANGUN SISTEM KONTROL PELONTAR PELURU DENGAN DUA SUMBU PUTAR

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat Memperoleh Gelar
Sarjana Teknik Mesin pada Bidang Studi Manufaktur

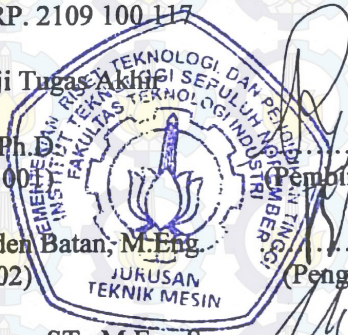
Program Studi S-1
Jurusan Teknik Mesin
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember Surabaya

Oleh :

IRVAN DWI PUTRA
NRP. 2109 100 117

Disetujui oleh Tim Penguji Tugas Akhir

1. Arif Wahyudi, ST., MT., Ph.D.
(NIP.19730322 200112 1 001) (Pembimbing)
2. Prof. Dr.Ing. I Made Londen Batan, M.Eng.
(NIP.195811061986011002) (Penguji 1)
3. Dr.Eng. Unggul Wasiwitono, ST., M.Eng.Sc.
(NIP.19780510 200112 1 001) (Penguji 2)
4. Ari Kurniawan Saputra ST, MT.
(NIP.....) (Penguji 3)



RANCANG BANGUN SISTEM KONTROL PELONTAR PELURU DENGAN DUA SUMBU PUTAR

Nama Mahasiswa : Irvan Dwi Putra
NRP : 2109 100 117
Jurusan : Teknik Mesin FTI-ITS
Dosen Pembimbing : Arif Wahjudi, ST., MT., Ph.D.

Abstrak

Perkembangan teknologi selaras dengan perkembangan jaman, salah satunya dibidang pertahanan. Tuntutan terhadap minimnya korban dalam penggunaan teknologi persenjataan menjadi hal utama yang ingin dicapai. Hal ini menuntut badan pertahanan suatu negara untuk selalu berinovasi dalam menciptakan suatu teknologi yang aman dan memiliki tingkat akurasi yang tinggi. Karena itu telah dilengkapi oleh pemerintah sebuah alat utama sistem pertahanan dengan persenjataan yang bersifat otomatis dan mampu dikendalikan dari jarak jauh. Senapan semi otomatis adalah salah satu alat persenjataan otomatis yang digunakan oleh badan pertahanan negara kita.

Dalam Tugas akhir ini dirancang dan dibuat sebuah prototype pelontar peluru yang dilengkapi dengan dua sumbu putar. Prototype sistem kontrol dirancang dengan bantuan software CodeVisionAVR, Software EAGLE dan diuji untuk menggerakan pelontar peluru sesuai dengan kordinat yang di input. Hasil program yang telah dirancang didownload ke dalam mikrokontroler menggunakan downloader. Mikrokontroler tersebut berfungsi menggerakan motor servo. Selain itu dibuat suatu sistem kontrol yang dapat menggerakan motor servo bergerak inklinasi $+60^\circ$ dan bergerak rotasi $\pm 60^\circ$ dengan presisi.

Dari Tugas Akhir ini didapatkan rancang bangun sistem kontrol pelontar peluru yang terdiri dari mikrokontroler hasil perakitan, dua motor servo dengan nilai torsi maksimal sebesar 3.7 kg.cm, dua potensiometer, pengatur relay, LCD pembaca nilai potensiometer. Selanjutnya dari percobaan yang dilakukan, sistem kontrol pelontar peluru dapat dikendalikan sesuai input yang dimasukan. Tetapi masih terjadi noise yang menyebabkan nilai toleransi kecermatan yang berbeda. Dimana nilai yang didapat berada diantara $\pm 0,38^\circ$ dan diluar $\pm 0,38^\circ$.

Kata kunci : pelontar peluru plastik, kestabilan sistem, unit kontrol, mikrokontroler

DESIGN OF BULLET LAUNCHER CONTROL SYSTEM WITH TWO-AXIS TURNTABLE

Student Name : Irvan Dwi Putra
NRP : 2109 100 117
Department : Teknik Mesin FTI-ITS
Academic Advisor : Arif Wahjudi, ST., MT., Ph.D.

Abstract

The development of technology in tune with the changing times, one of them in the field of Defense. Claim against its victims in the use of armaments technology became the main things you want to accomplish. It is demanding the country's Defense Agency to always innovate in creating a technology that safely and have a high degree of accuracy. Because it has been equipped by the Government's main defense system a tool with weaponry that is automated and able to be controlled remotely. Semi-automatic rifle is one of the tools of automatic weapons used by the defense of our country.

In this final project designed and built a prototype bullet Launcher is equipped with two rotary axes. Prototype control system designed with the help of software, Software CodeVisionAVR EAGLE and tested to drive the bullet Launcher corresponds to the coordinates in the input. The result of program that was designed to be downloaded into microcontroller using the downloader. The microcontroller function moves the servo motor. In addition to that created a system of control that can move the servo motor moves the inclination of $+ 60^\circ$ and $\pm 60^\circ$ rotation move with precision.

This final Task of mobilising design bullet launcher control system that consists of a microcontroller Assembly results, two servo motors with maximum torque rating of 3.7 kg.cm, two potentiometer, relay control, LCD reader value potentiometer. Next from experiments conducted, bullet launcher

control systems can be controlled according to the input being entered. But noise still occurs that causes the value of the tolerance accuracy. Where the value of the obtained between $\pm 0.38^\circ$ and $\pm 0.38^\circ$ outside.

Keywords: bullet Launcher, the stability of the system, the control unit, microcontroller.

KATA PENGANTAR

Assalamu'alaikum wr.wb.

Segala puji dan syukur penulis panjatkan kepada Allah SWT, Tuhan semesta alam yang selalu memberi kasih sayang tiada tara pada penulis sehingga mampu menyelesaikan Tugas Akhir dengan judul:

“RANCANG BANGUN SISTEM KONTROL PELONTAR PELURU DENGAN DUA SUMBU PUTAR”

Penulis menyadari bahwa dalam penyelesaian tugas akhir ini tidak lepas dari bantuan berbagai pihak. Untuk itu, pada kesempatan ini penulis menghaturkan ucapan terima kasih dan penghargaan setinggi-tingginya kepada :

1. Orangtua yang selalu mendoakan dan melakukan apapun demi senyum putra-putrinya.
2. Bapak Arif Wahjudi ST., MT., Ph.D. selaku dosen pembimbing Tugas Akhir, yang selalu sabar dan memberi penulis begitu banyak ilmu.
3. Bapak I Made Londen Batan, Bapak Unggul Wasiwitono dan Bapak Ari Kurniawan Saputra selaku dosen penguji Tugas Akhir, yang telah memberi ilmu dan pemahaman baru.
4. Bapak Yunarko T. selaku dosen wali yang telah membantu dan menyemangati penulis selama masa studi.
5. Adityo dan Alif Naufal partner dalam pembuatan Tugas Akhir Pelontar Peluru.
6. Kakak dan adik penulis, Sandra Kurnia Sari dan Muhammad Anggi yang selalu memberikan semangat dan motivasi.
7. Segenap Keluarga besar Lab. Perancangan dan Pengembangan Produk.

-
8. Teman-teman M-52 semua yang sudah memberikan saya waktu yang indah bersama kalian.
 9. Dosen dan karyawan Jurusan Teknik Mesin FTI-ITS, atas ilmu dan pengalaman serta bantuannya selama ini.
 10. Serta semua pihak yang tidak dapat penulis sebutkan satu per satu.

Akhirnya penulis menaruh sebuah harapan, semoga tugas akhir ini dapat memberikan manfaat bagi siapa saja yang membacanya.

Wassalamu'alaikum wr.wb.

Surabaya, 31 Juli 2015

Irvan Dwi Putra



DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	ii
ABSTRAK	iii
ABSTRACT	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
DAFTAR LAMPIRAN	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	2
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Manfaat Penelitian	3
BAB 2 TINJAUAN PUSTAKA	
2.1 Pengenalan Target Otomatis	5
2.2 Mikrokontroler	7
2.2.1 Arsitektur ATmega16	9
2.2.2 Konfigurasi Pena(Pin) ATmega16	10
2.2.3 Deskripsi Mikrokontroler ATmega16	11
2.2.4 Peta Memori ATmega16	12
2.2.5 <i>Analog To Digital Converter</i>	15
2.3 PWM	18
2.3.1 Jenis PWM	18
2.3.2 Cara Kerja dan Pengendalian	19
2.4 Motor servo	21
2.4.1 Jenis-Jenis Motor Servo	22
2.4.2 Pulsa Kontrol Motor Servo	22
2.4.3 Pulsa Kendali Motor Servo	22
2.5 Sensor Sudut	23
2.5.1 Potensiometer	24



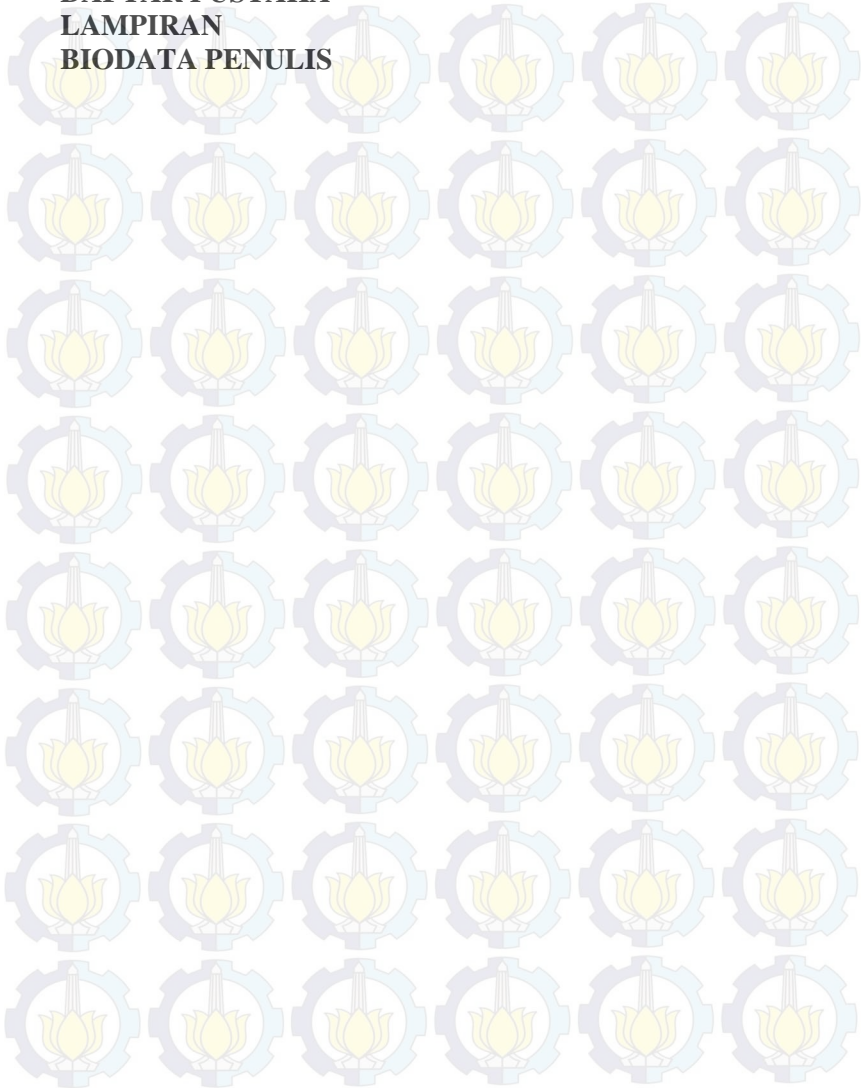
Tugas Akhir Manufaktur

2.6 CodeVisionAVR	26
2.7 Software EAGLE	27
2.8 Sistem PID	29
2.8.1 Kontrol Proporsional	30
2.8.2 Kontrol Integratif	32
2.8.3 Kontrol Derivatif	34
BAB 3 METODE PENELITIAN	37
3.1 Diagram Alir Penelitian	37
3.2 Langkah-Langkah Penelitian	38
3.3 Blok Diagram Sistem	40
BAB 4 PERANCANGAN DAN PERAKITAN	43
4.1 Spesifikasi Komponen	43
4.2 Perakitan (<i>Assembly</i>) Perangkat Keras	49
4.3 Resolusi Sistem	53
4.3.1 Resolusi Motor Servo	53
4.3.2 Resolusi Potensiometer	54
4.3.2 Kepresisian Sistem Kontrol	55
4.4 Algoritma Sistem	57
4.4.1 Algoritma Sistem Kontrol Pelontar Peluru ..	59
4.5 Perancangan <i>Interface</i> pada Microsoft Visual Basic 6.0	60
BAB 5 PENGUJIAN DAN ANALISA	63
5.1 Pengujian	63
5.2 Pengujian Perangkat keras (<i>Hardware</i>)	65
5.3 Analisa Hasil Pengujian	68
5.3.1 Analisa Hasil Pengujian Percobaan Pada Titik (-60° , 30°)	68
5.3.2 Analisa Hasil Pengujian Percobaan Pada Titik (0° , 30°)	71
5.3.3 Analisa Hasil Pengujian Percobaan Pada Titik ($+60^{\circ}$, 30°)	74
BAB 6 PENUTUP	79
6.1 Kesimpulan	79



6.2 Saran	80
-----------------	----

DAFTAR PUSTAKA
LAMPIRAN
BIODATA PENULIS





[Halaman Ini Sengaja Dikosongkan]



DAFTAR TABEL

Tabel 2.1 Konfigurasi <i>Clock</i> ADC	16
Tabel 2.2 Pemilihan Sumber Picu ADC	17
Tabel 4.1 Mikrokontroler ATmega 32	43
Tabel 4.2 Sistem Minimum	44
Tabel 4.3 USB to TTL Serial	44
Tabel 4.4 Motor Servo	45
Tabel 4.5 Relay 5 pin	45
Tabel 4.6 Potensiometer	46
Tabel 4.7 Adapter Listrik 2.5A	47
Tabel 4.8 Baterai	48
Tabel 4.9 Downloader USBasp	49
Tabel 5.1 Hasil Percobaan	68
Tabel 5.2 Tabel perhitungan Nilai rata-rata ($-60^{\circ}, 30^{\circ}$)	68
Tabel 5.3 Tabel perhitungan Nilai rata-rata ($0^{\circ}, 30^{\circ}$)	71
Tabel 5.4 Tabel perhitungan Nilai rata-rata ($+60^{\circ}, 30^{\circ}$)	74
Tabel 6.1 Komponen Sistem Kontrol Pelontar Peluru	79



[Halaman Ini Sengaja Dikosongkan]



DAFTAR GAMBAR

Gambar 2.1 Model sistem	7
Gambar 2.2 Blok Diagram Arsitektur ATmega16	9
Gambar 2.3 Pena-Pena Atmega16.....	10
Gambar 2.4 Peta Memori ATmega16	13
Gambar 2.5 Peta Memori Data ATmega16	14
Gambar 2.6 Control and Status Register A – ADCSRA	15
Gambar 2.7 ADC Multiplexer	16
Gambar 2.8 Register SFIOR	17
Gambar 2.9 Sinyal PWM	18
Gambar 2.10 Rangkaian PWM analog	18
Gambar 2.11 Sinyal PWM dan Persamaan Vout PWM	19
Gambar 2.12 Vrata-rata Sinyal PWM	20
Gambar 2.13 Duty Cycle dan Resolusi PWM	21
Gambar 2.14 Motor Servo	21
Gambar 2.15 Pulsa Kendali Motor Servo 180 ⁰	23
Gambar 2.16 Struktur Internal Potensiometer beserta bentuk dan simbol	25
Gambar 2.17 Potensiometer Putar	25
Gambar 2.18 Potensiometer Linear	26
Gambar 2.19 Tampilan umum <i>software CodeVisionAVR</i>	27
Gambar 2.20 <i>Schematic Editor</i>	27
Gambar 2.21 <i>Library Tool</i>	28
Gambar 2.22 <i>EAGLE Board</i>	28
Gambar 2.23 PID Blok Sistem	29
Gambar 2.24 Respon Output Kontroler P	31
Gambar 2.25 Respon Kontroler PI	33
Gambar 2.26 Respon Kontroler PID	35
Gambar 3.1 Diagram Alir Penelitian	38
Gambar 3.2 Blok Diagram Sistem	40
Gambar 4.1 Susunan Motor Servo arah Rotasi, Rumah Dudukan, dan <i>Fleksibel Joint</i>	49
Gambar 4.2 Susunan Motor Servo arah Inklinasi, Rumah Dudukan, Dan <i>Fleksibel Joint</i>	50

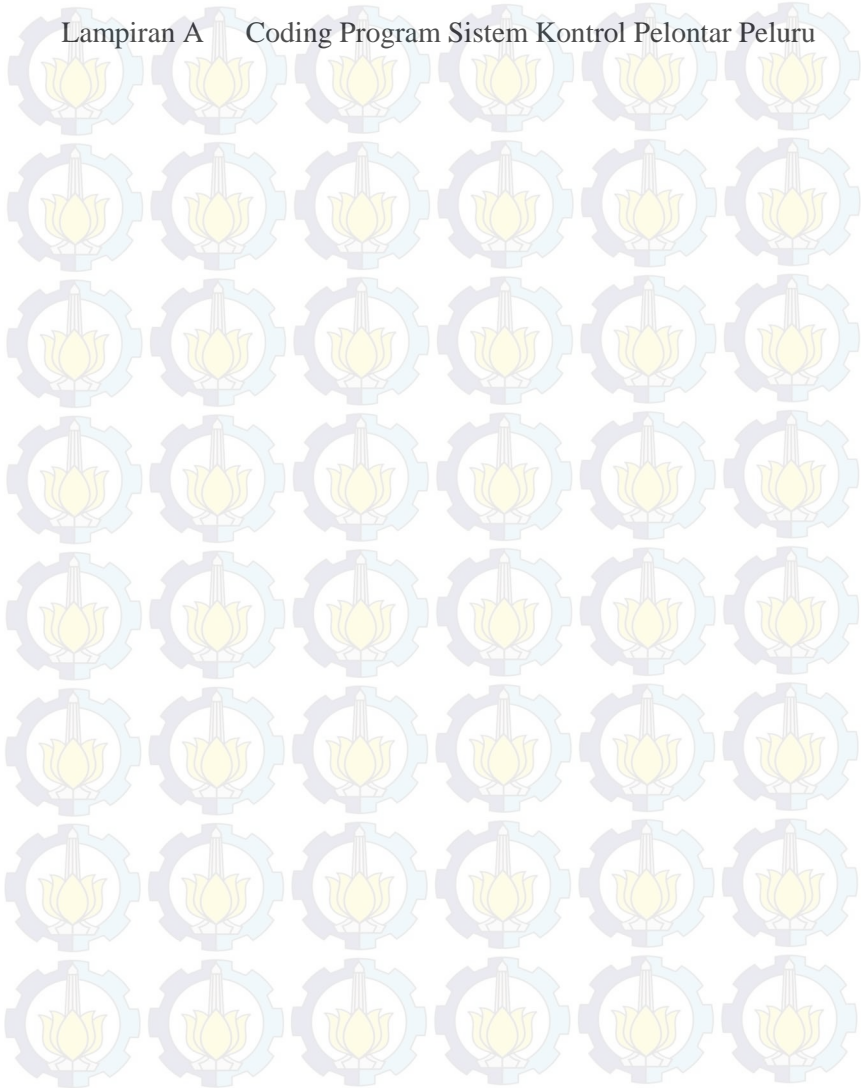


Gambar 4.3 USB To TTL Serial Terhubung Dengan Komputer	52
Gambar 4.4 Susunan <i>Hardware</i> Pelontar Peluru	53
Gambar 4.5 <i>flowchart</i> sistem kontrol pelontar peluru	60
Gambar 4.6 Interface pada Visual Basic 6.0	61
Gambar 5.1 Tampilan LCD	64
Gambar 5.2 Tes Input Manual Sudut	64
Gambar 5.3 <i>Interface</i> Program Pelontar Peluru	64
Gambar 5.4 Pengujian pertama Sudut ($-60^{\circ}, 30^{\circ}$)	65
Gambar 5.5 Nilai ADC dari percobaan sudut ($-60^{\circ}, 30^{\circ}$)	65
Gambar 5.6 Pengujian Sudut ($0^{\circ}, 30^{\circ}$)	66
Gambar 5.7 Nilai ADC dari percobaan sudut ($0^{\circ}, 30^{\circ}$)	66
Gambar 5.8 Pengujian Sudut ($+60^{\circ}, 30^{\circ}$)	66
Gambar 5.9 Nilai ADC dari percobaan sudut ($+60^{\circ}, 30^{\circ}$)	66
Gambar 5.10 Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik (-60°)	70
Gambar 5.11 Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik (30°)	70
Gambar 5.12 Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik (0°)	73
Gambar 5.13 Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik (30°)	73
Gambar 5.14 Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik ($+60^{\circ}$)	76
Gambar 5.15 Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik (30°)	76



DAFTAR LAMPIRAN

Lampiran A Coding Program Sistem Kontrol Pelontar Peluru





[Halaman Ini Sengaja Dikosongkan]



BAB I PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi selaras dengan perkembangan jaman, salah satunya dibidang pertahanan. Tuntutan terhadap minimnya korban dalam penggunaan teknologi persenjataan menjadi hal utama yang ingin dicapai. Hal ini menuntut badan pertahanan suatu negara untuk selalu berinovasi dalam menciptakan suatu teknologi yang aman dan memiliki tingkat akurasi yang tinggi. Oleh karena itu pemerintah telah melengkapi alat utama sistem pertahanan dengan persenjataan yang bersifat otomatis dan mampu dikendalikan dari jarak jauh. Senapan semi otomatis adalah salah satu alat persenjataan otomatis yang digunakan oleh badan pertahanan negara kita.

Saat ini telah banyak pengembangan senjata-senjata pengenalan target secara otomatis. Salah satu penelitian yang berhubungan dengan senjata pengenalan target secara otomatis [1]. Mereka meneliti berbagai macam aplikasi metode kontrol konvensional pada *automatic turret gun*. Selain itu terdapat penelitian tentang pengenalan target [2]. Mereka meneliti pemanfaatan sinyal *monogenic* untuk mendapatkan karakteristik pencitraan *synthetic aperture radar* (SAR) dengan teori *Dempster-Shafer* (DS). Dalam mengenali target melalui agregat informasi, memberikan hasil metode dalam metode ini yang jauh lebih efektif untuk menangani kondisi operasi yang berlebihan, seperti konfigurasi yang berbeda dan depresi. Selain itu paper tentang sistem kontrol otomatis pada robot telah dikembangkan [3]. Sistem *ballast* yang dikembangkan dikontrol dengan penghantar kabel sehingga masukan yang diterima oleh mikrokontroler. Robot bawah air dapat menyelam di dalam air sesuai kedalaman yang diinginkan dan mengapung ke kondisi awal dengan menggunakan sistem *ballast*.

Dari beberapa penelitian yang telah ada, kita sebagai warga Negara Indonesia harus berperan aktif dalam membantu



setiap pembangunan. Salah satunya adalah pertahanan Negara. Dengan membantu menciptakan alat senjata pengenalan otomatis yang sudah berkembang. Setiap persenjataan harus memiliki tingkat akurasi dan kepresisian. Agar rancangan ini sesuai dengan kebutuhan bidang pertahanan, maka dibuat suatu pemodelan *prototype* dengan menggunakan pelontar peluru plastik. Dari *prototype* ini dilakukan pengujian perancangan sistem kontrol dengan bantuan *software CodeVisionAVR* pada laptop untuk menggerakkan pelontar peluru plastik sesuai dengan kordinat yang kita input. Hasil program yang telah dirancang didownload ke dalam mikrokontroler menggunakan downloader. Mikrokontroler tersebut yang akan menggerakkan motor servo. Pada tugas akhir ini akan dibuat suatu sistem kontrol yang dapat menggerakkan motor servo bergerak naik, ke kiri dan ke kanan dengan cepat, sehingga dapat mempercepat proses kerja motor servo tersebut.

1.2 Rumusan Masalah

Rumusan masalah dari penulisan laporan tugas akhir ini adalah:

- Bagaimanakah perancangan sistem kontrol pelontar peluru plastik yang mempunyai dua sumbu putar?
- Bagaimana membangun sistem kontrol pelontar peluru plastik dengan kecermatan $\pm 0.38^\circ$?

1.3 Tujuan

Tujuan dari penulisan Tugas Akhir ini adalah:

- Perancangan sistem kontrol pelontar peluru plastik dengan dua sumbu putar.
- Membangun sistem kontrol pelontar peluru plastik dengan toleransi kecermatan $\pm 0.38^\circ$.

1.4 Batasan Masalah

Batasan masalah dari penulisan tugas akhir ini adalah sebagai berikut:

- Jarak inklinasi $+60^\circ$ pada motor servo 1.



- b. Jarak rotasi $\pm 60^\circ$ pada motor servo 2.
- c. Menggunakan bantuan software *CodeVisionAVR*, *EAGLE* dan *Visual Basic*.
- d. Menggunakan sistem kontrol *ATMega32*.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat pada semua pihak. Manfaat dari penelitian ini adalah:

- a. Memperkaya dan memperdalam wacana dalam bidang ilmu sistem kontrol, terutama pada perancangan sistem kontrol pelontar peluru.
- b. Memberikan kontribusi nyata pada pengembangan teknologi perancangan pelontar peluru, demi terciptanya kemajuan teknologi nasional khususnya di bidang pertahanan.
- c. Mesin pelontar peluru yang telah berfungsi dapat digunakan untuk penelitian lebih lanjut.



[Halaman Ini Sengaja Dikosongkan]



BAB II

TINJAUAN PUSTAKA

2.1 Pengenalan Target Otomatis

Salah satu penelitian yang berhubungan dengan senjata adalah pengenalan target secara otomatis [1]. Mereka meneliti berbagai macam aplikasi metode kontrol konvensional pada *automatic turret gun*. Seperti kontrol *Proportional-Integral-Derivative controller* (PID), *Resolved Acceleration Control* (RAC) dan *Resolved Acceleration Control Active Force Control* (RACAFC). Sejumlah percobaan telah dilakukan untuk berbagai macam sudut baik di *azimuth* dan elevasi sumbu *turret gun*. Dari hasil penelitian menunjukkan bahwa untuk sistem ATG yang bekerja dibawah gangguan, RACAFC menunjukkan performa yang lebih baik dari kedua metode kontrol, yaitu RAC dan PID. Akan tetapi jika percobaan tanpa pemberian beban, hasil yang memuaskan didapat dari RAC dan RACAFC, sedangkan pada kontrol PID tidak. Dikarenakan kontrol PID tidak dapat menjangkau semua sudut yang diberikan. Selain itu terdapat penelitian tentang pengenalan target [2]. Penelitian yang dilakukan adalah pemanfaatan sinyal monogenic untuk mendapatkan karakteristik pencitraan *synthetic aperture radar* (SAR) dengan teori *Dempster-Shafer* (DS). Dalam mengenali target melalui agregat informasi, memberikan hasil metode dalam metode ini yang jauh lebih efektif untuk menangani kondisi operasi yang berlebihan, seperti konfigurasi yang berbeda dan depresi.

Paper lain [4] menjelaskan perencanaan dan desain dari semi-otonom robot penjaga (menara) dengan menerapkan biaya rendah pada pembuatanya. Robot ini memiliki senjata *Nerfgun* otomatis yang dipasang pada atas *platform*. Sensor yang digunakan pengenalan target sasaran adalah *Sharp IR proximity* sensor merupakan sensor jarak inframerah. Robot dapat diatur ke mode penjaga otonom, dimana ia dapat melacak target di sekitar mereka secara otomatis, atau bisa juga dengan modus pengendalian, untuk dikendalikan ke lokasi yang baru.



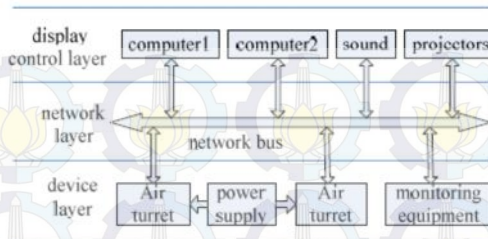
Konsep dari penggunaan sensor *panning* IR merupakan konsep yang sederhana. Pertama, untuk mengontrol robot harus dipilih sebuah mikrokontroler yang tepat. Dalam pembuatan robot ini digunakan mikrokontroler Arduino Pro digabungkan dengan Atmel AVR Atmega 1280. Dikarenakan Arduino cukup mudah untuk digunakan sebab didalam arduino sudah terdapat beberapa komponen yang sudah tertanam, seperti ADC untuk IR sensor, PWM untuk mengendalikan servo, dan USART yang untuk berkomunikasi dengan remote kontrol dan motor. Pemilihan mikrokontroler haruslah disesuaikan oleh beberapa hal yaitu kebutuhan, kemampuan *hardware*, dan juga dukungan *Software* yang akan digunakan.

Kesimpulannya dari makalah ini didapatkan kinerja robot yang lebih baik dibandingkan dengan desain murah lainnya dengan penggunaan *hardware* yang sama, tetapi kinerjanya sebanding dengan desain *hardware* dan sensor mahal lainnya.

Pada paper [5] ini dijelaskan pemodelan dan simulasi untuk pengebom pada menara udara. Metode yang digunakan untuk pengenalan target dengan *virtual flight* modul melalui data pemandangan penerbangan, dan sasaran yang telah disimpan pada komputer untuk selanjutnya data tersebut akan diolah pada komputer sebagai pengenalan target.

Desain Sistem ini merupakan sistem simulasi untuk diterapkan dalam praktik mengajar dan live-amunisi simulasi pelatihan. banyak hal yang dapat dilakukan seperti konfigurasi, pemilihan target, pemilihan lingkungan dan lain sebagainya. Alat ini dapat mensimulasikan sasaran, proyektil, target dan sasaran ledakan. Pada saat yang sama, dapat memonitor stabilitas sistem, Analisis aero menembak menara akurasi, simulasi data dan menilai hasil tembakan untuk menyimpan hasil.

Sistem ini terdiri dari menara udara, komputer, pasokan listrik, suara, proyektor, printer, hub dan lain - lain dapat dibagi menjadi tiga lapisan, masing-masing menampilkan lapisan kontrol, jaringan lapisan dan lapisan perangkat, seperti ditunjukkan pada Gambar 2.1 model sistem.



Gambar 2.1 Model sistem [5]

Lapisan ini terdiri dari dua komputer berkecepatan tinggi, proyektor dan peralatan sound. Komputer utama dilengkapi dengan *Software* simulasi adalah bagian inti dari keseluruhan sistem. Digunakan untuk menyelesaikan operasi antarmuka pengguna dan penseleksian target.

Berdasarkan studi struktur menara bomber penerbangan, dengan dibuatnya sistem simulasi pelatihan terpadu yang dapat dibuat sebagai tempat berlatih pemilihan target, seleksi dan sebagainya, sehingga dapat mensimulasikan visual, target, memonitor stabilitas dan menganalisis ketepatan menembak. Ini merupakan metode yang efektif untuk peserta pelatihan agar akrab dengan struktur *gun* pesawat dan metode penggunaannya, sehingga dapat membantu peserta untuk mengoperasikan senjata pesawat.

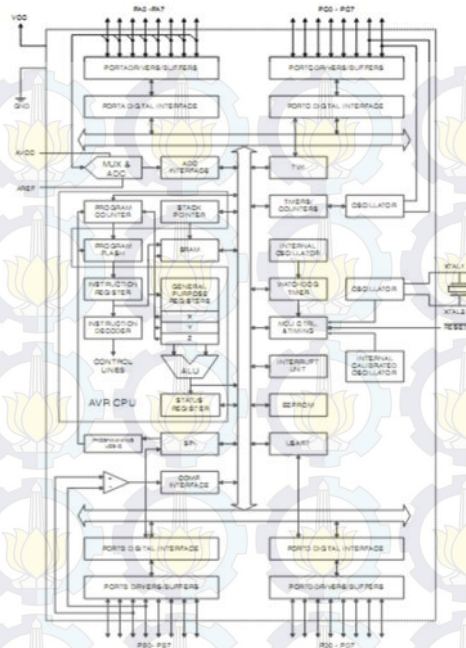
2.2 Mikrokontroler

Mikrokontroler [6] adalah suatu piranti yang digunakan untuk mengolah data-data biner (*digital*) yang didalamnya merupakan gabungan dari rangkaian-rangkaian elektronik yang dikemas dalam bentuk suatu *chip* (IC). Mikrokontroler adalah suatu keping IC dimana terdapat mikroprosesor dan memori program (ROM) serta memori serbaguna (RAM), bahkan ada beberapa jenis mikrokontroler yang memiliki fasilitas ADC, PLL, EEPROM dalam satu kemasan. Penggunaan mikrokontroler dalam bidang kontrol sangat luas dan populer. Mikrokontroler merupakan komputer didalam *chip* yang digunakan untuk mengontrol peralatan elektronik,



yang menekankan efisiensi dan efektifitas biaya. Secara harfiahnya bisa disebut “pengendali kecil” dimana sebuah sistem elektronik yang sebelumnya banyak memerlukan komponen-komponen pendukung seperti IC, TTL dan CMOS dapat direduksi/diperkecil dan akhirnya terpusat serta dikendalikan oleh mikrokontroler ini.

Pada umumnya mikrokontroler terdiri dari bagian-bagian sebagai berikut: alamat (*address*), data, pengendali, memori (RAM atau ROM), dan bagian *input-output*. AVR merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur *Reduced Instruction Set 16 Computer* (RISC). Hampir semua instruksi dieksekusi dalam satu siklus clock. AVR mempunyai 32 register *general purpose*, *timer/counter* fleksibel dengan *mode compare*, *interrupt* internal dan eksternal, serial UART, *programmable Watchdog Timer*, dan *mode power saving*. Mempunyai ADC dan PWM internal. AVR juga mempunyai *In-System Programmable Flash on-chip* yang memungkinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI. ATmega16 adalah mikrokontroler CMOS 8-bit daya-rendah berbasis arsitektur RISC yang ditingkatkan. Gambar 2.2 merupakan gambar dari blok diagram arsitektur Atmega16.



Gambar 2.2 Blok Diagram Arsitektur ATmega16 [13]

2.2.1 Arsitektur ATMEGA16

Mikrokontroler ini menggunakan arsitektur Harvard yang memisahkan memori program dari memori data, baik bus alamat maupun bus data, sehingga pengaksesan program dan data dapat dilakukan secara bersamaan (*concurrent*). Secara garis besar mikrokontroler ATmega16 [14] terdiri dari :

1. Arsitektur RISC dengan *throughput* mencapai 16 MIPS pada frekuensi 16Mhz.
2. Memiliki kapasitas *flash* memori 16 Kbyte, EEPROM 512 Byte, dan SRAM 1 Kbyte.
3. Saluran I/O 32 buah, yaitu Bandar A, Bandar B, Bandar C, dan Bandar D.
4. CPU yang terdiri dari 32 buah register.
5. U ser interupsi internal dan eksternal.



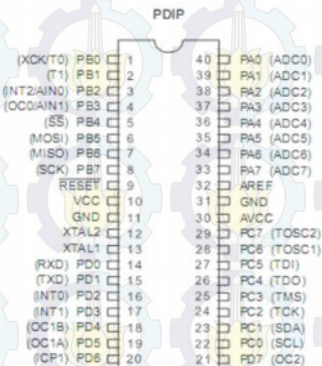
6. Bandar antarmuka SPI dan Bandar USART sebagai komunikasi serial.

7. Fitur Peripheral.

- Dua buah 8-bit *timer/counter* dengan prescaler terpisah dan mode *compare*.
- Satu buah 16-bit *timer/counter* dengan prescaler terpisah, mode *compare*, dan mode *capture*.
- *Real time counter* dengan osilator tersendiri.
- Empat kanal PWM dan Antarmuka komparator analog.
- 8 kanal, 10 bit ADC.
- *Byte-oriented Two-wire Serial Interface*.
- *Watchdog timer* dengan osilator internal.

2.2.2 KONFIGURASI PENA (PIN) ATMEGA16

Konfigurasi pena (*pin*) mikrokontroler Atmega16 dengan kemasan 40 pena dapat dilihat pada Gambar 2.3. Dari gambar tersebut dapat terlihat ATmega16 memiliki 8 pena untuk masing-masing bandar A (*Port A*), bandar B (*Port B*), bandar C (*Port C*) dan bandar D (*Port D*).



Gambar 2.3 Pena-Pena Atmega16 [13]



2.2.3 DESKRIPSI MIKROKONTROLER ATMEGA16

- VCC (*Power Supply*) dan GND (*Ground*)
- Bandar A (PA7..PA0)

Bandar A berfungsi sebagai *input* analog pada konverter A/D. Bandar A juga sebagai suatu bandar I/O 8-bit dua arah, jika A/D konverter tidak digunakan. Pena-pena Bandar dapat menyediakan resistor *internal pull-up* (yang dipilih untuk masing-masing bit). Bandar A *output* buffer mempunyai karakteristik gerakan simetris dengan keduanya sink tinggi dan kemampuan sumber. Ketika pena PA0 ke PA7 digunakan sebagai input dan secara eksternal ditarik rendah, pena-pena akan memungkinkan arus sumber jika resistor *internal pull-up* diaktifkan. Pena Bandar A adalah *tri-stated* manakala suatu kondisi reset menjadi aktif, sekalipun waktu habis.
- Bandar B (PB7..PB0)

Bandar B adalah suatu bandar I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). Bandar B *output buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Sebagai input, pena Bandar B yang secara eksternal ditarik rendah akan arus sumber jika resistor *pull-up* diaktifkan. Pena Bandar B adalah *tri-stated* manakala suatu kondisi reset menjadi aktif, sekalipun waktu habis.
- Bandar C (PC7..PC0)

Bandar C adalah suatu bandar I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). Bandar C *output buffer* mempunyai karakteristik gerakan simetris dengan keduanya sink tinggi dan kemampuan sumber. Sebagai input, pena bandar C yang secara eksternal ditarik rendah akan arus sumber jika resistor *pull-up* diaktifkan. Pena bandar C adalah *tri-stated*



manakala suatu kondisi reset menjadi aktif, sekalipun waktu habis.

- Bandar D (PD7..PD0)

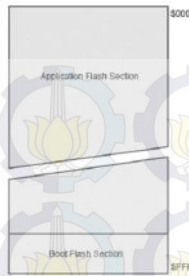
Bandar D adalah suatu bandar I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). Bandar D *output buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Sebagai input, pena bandar D yang secara eksternal ditarik rendah akan arus sumber jika resistor *pull-up* diaktifkan. Pena Bandar D adalah *tri-stated* manakala suatu kondisi reset menjadi aktif, sekalipun waktu habis.

- RESET (*Reset input*)
- XTAL1 (*Input Oscillator*)
- XTAL2 (*Output Oscillator*)
- AVCC adalah pena penyedia tegangan untuk bandar A dan Konverter A/D.
- AREF adalah pena referensi analog untuk konverter A/D.

2.2.4 Peta Memori ATMega16

2.2.4.1. Memori Program

Arsitektur ATMega16 mempunyai dua memori utama, yaitu memori data dan memori program. Selain itu, ATMega16 memiliki memori EEPROM untuk menyimpan data. ATMega16 memiliki 16 Kbyte *On-chip In-System Reprogrammable Flash Memory* untuk menyimpan program. Instruksi ATMega16 semuanya memiliki format 16 atau 32 bit, maka memori *flash* diatur dalam 8K x 16 bit. Memori *flash* dibagi kedalam dua bagian, yaitu bagian program *boot* dan aplikasi seperti terlihat pada Gambar 2.4. *Bootloader* adalah program kecil yang bekerja pada saat sistem dimulai yang dapat memasukkan seluruh program aplikasi ke dalam memori prosesor.



Gambar 2.4 Peta Memori ATMega16 [13]

2.2.4.2 Memori Data (SRAM)

Memori data AVR ATMega16 pada gambar 2.5 terbagi menjadi 3 bagian, yaitu 32 register umum, 64 buah register I/O dan 1 Kbyte SRAM internal. *General purpose register* menempati alamat data terbawah, yaitu \$00 sampai \$1F. Sedangkan memori I/O menempati 64 alamat berikutnya mulai dari \$20 hingga \$5F. Memori I/O merupakan register yang khusus digunakan untuk mengatur fungsi terhadap berbagai fitur mikrokontroler seperti kontrol *register*, *timer/counter*, fungsi-fungsi I/O, dan sebagainya. 1024 alamat berikutnya mulai dari \$60 hingga \$45F digunakan untuk SRAM internal.

Register File

Data Address Space

R0	\$0000
R1	\$0001
R2	\$0002
...	...
R29	\$000D
R30	\$000E
R31	\$000F



I/O Registers

\$00	\$0020
\$00	\$0021
\$01	\$0022
...	...
\$3D	\$005D
\$3E	\$005E
\$3F	\$005F

Internal SRAM

\$0061
...
\$045E
\$045F

Gambar 2.5 Peta Memori Data ATmega16 [13]

2.2.4.3 Memori Data EEPROM

ATmega16 terdiri dari 512 byte memori data EEPROM 8 bit, data dapat ditulis/dibaca dari memori ini, ketika catu daya dimatikan, data terakhir yang ditulis pada memori EEPROM masih tersimpan pada memori ini, atau dengan kata lain memori EEPROM bersifat *nonvolatile*. Alamat EEPROM mulai dari \$000 sampai \$1FF.



2.2.5 Analog To Digital Converter

AVR ATmega16 merupakan tipe AVR yang telah dilengkapi dengan 8 saluran ADC internal dengan resolusi 10 bit. Dalam mode operasinya, ADC dapat dikonfigurasi, baik *single ended* input maupun *differential* input. Selain itu, ADC ATmega16 memiliki konfigurasi pewaktuan, tegangan referensi, mode operasi, dan kemampuan filter derau (*noise*) yang amat fleksibel sehingga dapat dengan mudah disesuaikan dengan kebutuhan dari ADC itu sendiri. ADC pada ATmega16 memiliki fitur-fitur antara lain :

- Resolusi mencapai 10-bit.
- Akurasi mencapai ± 2 LSB.
- Waktu konversi 13-260 μ s.
- 8 saluran ADC dapat digunakan secara bergantian.
- Jangkauan tegangan input ADC bernilai dari 0 hingga VCC.
- Disediakan 2,56V tegangan referensi internal ADC.
- Mode konversi kontinyu atau mode konversi tunggal.
- Interupsi ADC *complete*.
- *Sleep Mode Noise canceler*.

Proses inisialisasi ADC meliputi proses penentuan *clock*, tegangan referensi, format data keluaran, dan modus pembacaan. Register-register yang perlu diatur adalah sebagai berikut:

- ADC Control and Status Register A – ADCSRA

Bit	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.6 ADC Control and Status Register A – ADCSRA [13]

ADEN : 1 = adc *enable*, 0 = adc *disable*

ADCS : 1 = mulai konversi, 0 = konversi belum terjadi



ADATE : 1 = *auto trigger* diaktifkan, *trigger* berasal dari sinyal yang dipilih (set pada *trigger* SFIOR bit ADTS). ADC akan start konversi pada edge positif sinyal *trigger*.

ADIF : diset ke 1, jika konversi ADC selesai dan data *register* ter-update. Namun ADC *Conversion Complete Interrupt* dieksekusi jika bit ADIE dan bit-I dalam *register* SREG diset.

ADIE : diset 1, jika bit-I dalam *register* SREG di-set.

ADPS[0..2] : Bit pengatur clock ADC, faktor pembagi 0 ... 7
= 2, 4, 8, 16, 32, 64, 128.

Tabel 2.1 Konfigurasi *Clock* ADC

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

- ADC Multiplexer-ADMUX

Bit	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.7 ADC Multiplexer [13]

REFS 0, 1 : Pemilihan tegangan referensi ADC

00 : Vref = Aref

01 : vref = AVCC dengan eksternal *capasitor* pada AREF



10 : vref = internal 2.56 volt dengan eksternal kapasitor pada AREF

ADLAR : Untuk *setting* format data hasil konversi ADC, default = 0

- *Special Function IO Register-SFIOR*

SFIOR merupakan *register* 8 bit pengatur sumber picu konversi ADC, apakah dari picu eksternal atau dari picu internal, susunannya seperti yang terlihat pada Gambar 2.8 berikut:



Gambar 2.8 Register SFIOR [13]

ADTS[0...2] : Pemilihan *trigger* (pengatur picu) untuk konversi ADC, bit-bit ini akan berfungsi jika bit ADATE pada *register* ADCSRA bernilai 1. Konfigurasi bit ADTS[0...2] dapat dilihat pada Tabel 2.2.

Tabel 2.2 Pemilihan sumber picu ADC

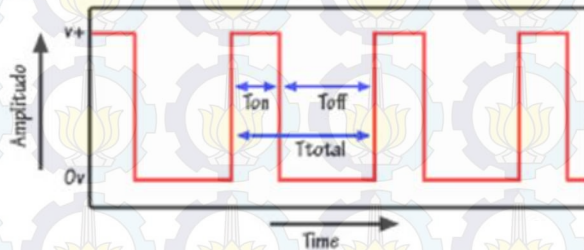
ADTS2	ADTS1	ADTS0	Trigger Source
0	0	0	Free Running mode
0	0	1	Analog Comparator
0	1	0	External Interrupt Request 0
0	1	1	Timer/Counter0 Compare Match
1	0	0	Timer/Counter0 Overflow
1	0	1	Timer/Counter Compare Match B
1	1	0	Timer/Counter1 Overflow
1	1	1	Timer/Counter1 Capture Event

ADHSM : 1. ADC *high speed mode enabled*. Untuk operasi ADC, bit ACME, PUD, PSR2 dan PSR10 tidak diaktifkan.



2.3 PWM

Pulse Width Modulation (PWM) secara umum adalah sebuah cara memanipulasi lebar sinyal yang dinyatakan dengan pulsa dalam suatu perioda, untuk mendapatkan tegangan rata-rata yang berbeda. Beberapa Contoh aplikasi PWM adalah pemodulasian data untuk telekomunikasi, pengontrolan daya atau tegangan yang masuk ke beban, regulator tegangan, *audio effect* dan penguatan, serta aplikasi-aplikasi lainnya. Sinyal PWM ditunjukkan pada gambar 2.9.



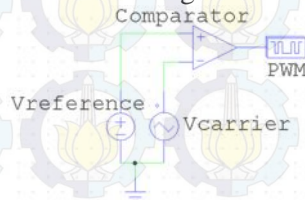
Gambar 2.9 Sinyal PWM [11]

Aplikasi PWM berbasis mikrokontroler biasanya berupa, pengendalian kecepatan motor DC, pengendalian motor servo dan pengaturan nyala terang LED.

2.3.1 Jenis PWM

2.3.1.1 Analog

Pembangkitan sinyal PWM yang paling sederhana adalah dengan cara membandingkan sinyal gigi gergaji sebagai tegangan *carrier* dengan tegangan referensi menggunakan rangkaian op-amp *comparator*. Rangkaian PWM analog bisa dilihat pada gambar 2.10.



Gambar 2.10 Rangkaian PWM analog [12]

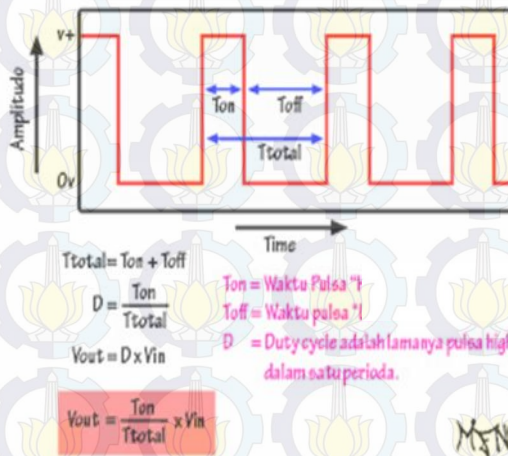


2.3.1.2 Digital

Pada metode digital setiap perubahan PWM dipengaruhi oleh resolusi dari PWM itu sendiri. Misalkan PWM *digital* 8 bit berarti PWM tersebut memiliki resolusi $2^8 = 256$, maksudnya nilai keluaran PWM ini memiliki 256 variasi, variasinya mulai dari 0 – 255 yang mewakili duty cycle 0 – 100% dari keluaran PWM tersebut.

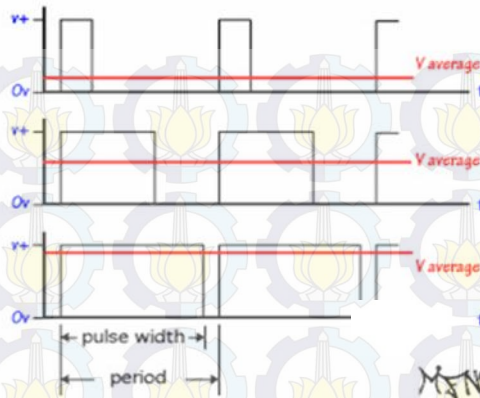
2.3.1 Cara Kerja dan Pengendalian

Sinyal PWM pada umumnya memiliki amplitudo dan frekuensi dasar yang tetap, namun memiliki lebar pulsa yang bervariasi. Lebar Pulsa PWM berbanding lurus dengan amplitudo sinyal asli yang belum termodulasi. Artinya, Sinyal PWM memiliki frekuensi gelombang yang tetap namun *duty cycle* bervariasi (antara 0% hingga 100%). Sinyal PWM dan Persamaan V_{out} PWM bisa dilihat pada Gambar 2.11.



Gambar 2.11 Sinyal PWM dan Persamaan V_{out} PWM [11]

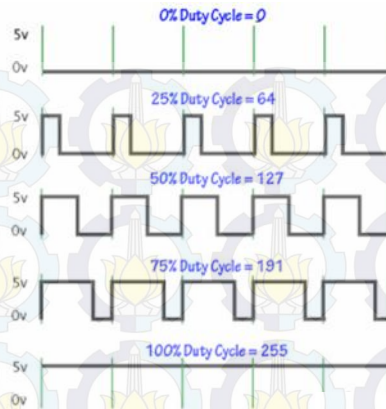
Dari persamaan diketahui bahwa perubahan *duty cycle* akan merubah tegangan keluaran atau tegangan rata-rata seperti gambar 2.12.



Gambar 2.12 Vrata-rata Sinyal PWM [11]

Pulse Width Modulation (PWM) merupakan salah satu teknik untuk didapatkan *signal* analog dari sebuah piranti *digital*. Sebenarnya Sinyal PWM dapat dibangkitkan dengan banyak cara, dapat menggunakan metode analog dengan menggunakan rangkaian op-amp atau dengan menggunakan metode *digital*.

Dengan metode analog setiap perubahan PWM-nya sangat halus, sedangkan menggunakan metode digital setiap perubahan PWM dipengaruhi oleh resolusi dari PWM itu sendiri. Resolusi adalah jumlah variasi perubahan nilai dalam PWM tersebut. Misalkan suatu PWM memiliki resolusi 8 bit berarti PWM ini memiliki variasi perubahan nilai sebanyak $2^8 = 256$ variasi mulai dari 0 – 255 perubahan nilai yang mewakili duty cycle 0 – 100% dari keluaran PWM tersebut. *Duty Cycle* dan Resolusi PWM bisa dilihat pada gambar 2.13.

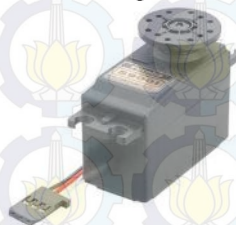


Gambar 2.13 Duty Cycle dan Resolusi PWM [12]

2.4 Motor Servo

Motor Servo [7] adalah sebuah motor DC yang dilengkapi rangkaian kendali dengan system *closed feedback* yang terintegrasi dalam motor tersebut. Pada motor servo posisi putaran sumbu (*axis*) dari motor akan diinformasikan kembali ke rangkaian control yang ada di dalam motor servo.

Motor servo disusun dari sebuah motor DC, *gearbox*, *variable resistor* (VR) atau potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas maksimum putaran sumbu (*axis*) motor servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang ada pada pin kontrol motor servo. Salah satu jenis motor servo pada Gambar 2.14.



Gambar 2.14 Motor Servo [7]



2.4.1 Jenis-Jenis Motor Servo

2.4.1.1 Motor Servo Standar 180°

Motor servo jenis ini hanya mampu bergerak dua arah (CW dan CCW) dengan defleksi sudut putar (dapat berputar secara kontinu).

2.4.1.2 Motor Servo *Continuous*

Motor servo jenis ini mampu bergerak dua arah (CW dan CCW) tanpa batasan defleksi sudut putar (dapat berputar secara kontinu).

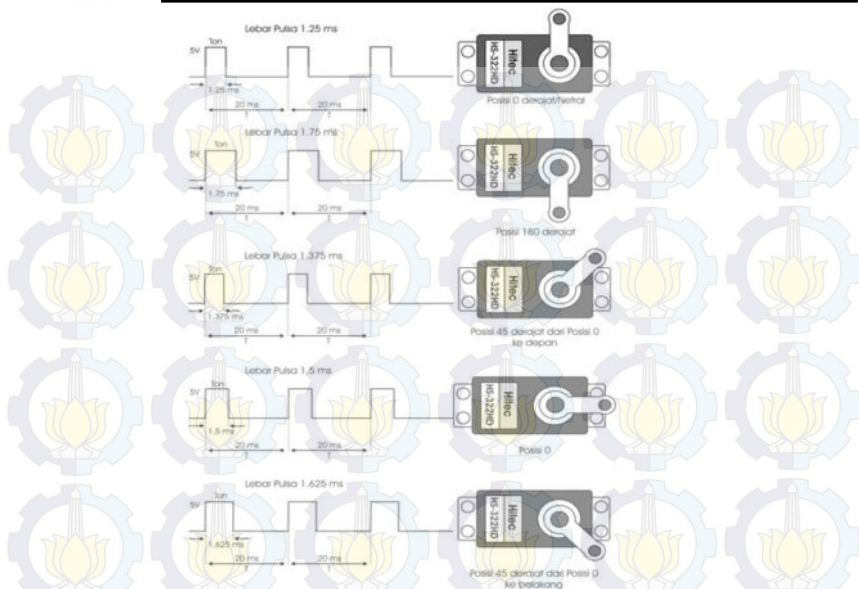
2.4.2 Pulsa Kontrol Motor Servo

Operasional motor servo dikendalikan oleh sebuah pulsa selebar ± 20 ms, dimana lebar pulsa antara 0.5 ms dan 2 ms menyatakan akhir dari range sudut maksimum. Apabila motor servo diberikan pulsa sebesar 1.5 ms mencapai gerakan 90°, maka bila kita berikan pulsa kurang dari 1.5 ms maka posisi mendekati 0° dan bila kita berikan pulsa lebih dari 1.5 ms maka posisi mendekati 180°.

2.4.3 Pulsa Kendali Motor Servo

Motor servo akan bekerja secara baik jika pada bagian pin kontrolnya diberikan sinyal PWM dengan frekuensi 50 Hz. Dimana pada saat sinyal dengan frekuensi 50 Hz tersebut dicapai pada kondisi $T_{on} \text{ duty cycle}$ 1.5 ms, maka rotor dari motor akan berhenti tepat di tengah-tengah (sudut 0° atau netral).

Pada saat $T_{on} \text{ duty cycle}$ dari sinyal diberikan kurang dari 1.5 ms, maka rotor akan berputar berlawanan arah jarum jam *Counter Clock Wise* (CCW) dengan membentuk sudut yang besarnya linear terhadap besarnya $T_{on} \text{ duty cycle}$ dan akan bertahan diposisi tersebut. Dan sebaliknya, jika $T_{on} \text{ duty cycle}$ dari sinyal yang diberikan lebih dari 1.5 ms, maka rotor akan berputar searah jarum jam *Clock Wise* (CW) dengan membentuk sudut yang linear pula terhadap besarnya $ton \text{ duty cycle}$ dan bertahan diposisi tersebut. Penjelasan bisa dilihat pada Gambar 2.15 Pulsa kendali motor servo 180°.



Gambar 2.15 Pulsa Kendali Motor Servo 180° [7]

2.5 Sensor Sudut

Sensor adalah peranti atau *device* yang digunakan untuk mengindera (*to sense*) besaran fisik. Sensor terdiri dari transduser, yaitu alat yang mengubah energi dari satu bentuk ke bentuk yang lain. Transduser dapat dibedakan menjadi dua macam yaitu transduser *input* dan transduser *output*. Transduser *input* mengubah energi non listrik menjadi energi listrik. Sedangkan transduser *output* adalah kebalikannya, mengubah energi listrik menjadi energi non listrik. Secara umum sensor dapat dibedakan menjadi beberapa macam sesuai prinsip kerja dan besaran ukurnya. Antara lain sensor kedekatan (*proximity*), sensor suhu, sensor sinar, sensor ultrasonik, sensor kecepatan, dan lain sebagainya.

Sensor sinar (cahaya) adalah jenis sensor yang mendeteksi perubahan, pantulan, atau bias cahaya yang berasal dari sumber cahaya dan mengenai benda atau ruangan. Sensor sinar mengubah



energi sinar langsung menjadi energi listrik. Sensor jenis ini tersiri dari 3 kategori:

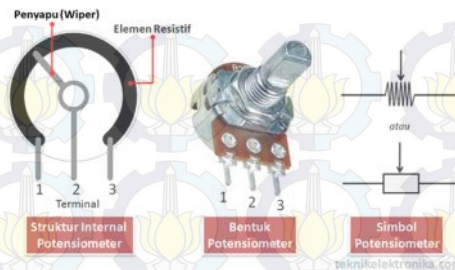
1. **Fotovoltaic** atau sel solar memiliki prinsip kerja adanya penyinaran cahaya menyebabkan pergerakan elektron dan menghasilkan tegangan.
2. **Fotokonduktif** atau fotoresistif memiliki prinsip kerja perubahan tahanan (resistansi) pada sel-sel ketika dikenai cahaya. Semakin besar intensitas cahaya maka nilai tahanan menjadi kecil. Dan sebaliknya, semakin kecil intensitas cahaya, nilai tahanan yang dihasilkan akan semakin besar.
3. **Fotolistrik** berprinsip kerja berdasarkan pantulan karena perubahan jarak atau posisi suatu sumber sinar (inframerah atau laser) dan bidang pemantulnya.

2.5.1 Potensiometer

Potensiometer adalah resistor tiga terminal dengan sambungan geser yang membentuk pembagi tegangan dapat disetel [8]. Jika hanya dua terminal yang digunakan (salah satu terminal tetap dan terminal geser), potensiometer berperan sebagai resistor variabel atau Rheostat. Potensiometer biasanya digunakan untuk mengendalikan peranti elektronik seperti pengendali suara pada penguat. Potensiometer yang dioperasikan oleh suatu mekanisme dapat digunakan sebagai transduser, misalnya sebagai sensor *joystick*. Secara struktur, Potensiometer terdiri dari 3 kaki Terminal dengan sebuah *shaft* atau tuas yang berfungsi sebagai pengaturnya. Gambar 2.16 menunjukkan Struktur Internal Potensiometer beserta bentuk dan Simbolnya.



POTENSIOMETER

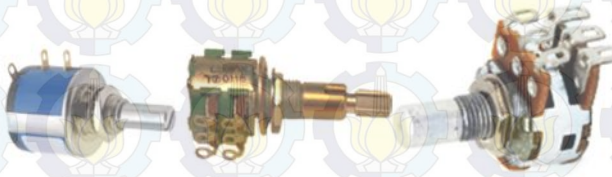


Gambar 2.16 Struktur Internal Potensiometer beserta bentuk dan simbol [8]

2.5.1.1 Jenis potensiometer

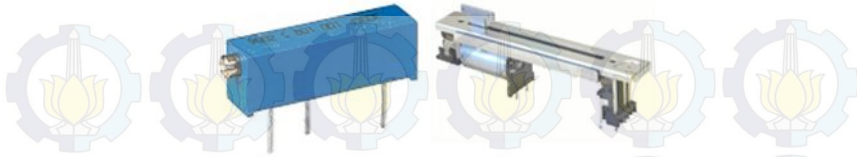
1. Secara manual potensiometer dapat dibagi menjadi 2 jenis, yaitu potensiometer dengan gerakan berputar (potensior putar) dan potensiometer linier.

- Potensiometer putar adalah jenis potensiometer yang paling umum dimana *wiper* bergerak dengan jalan melingkar (memutar), seperti pada gambar 2.17.



Gambar 2.17 Potensiometer Putar [8]

- Potensiometer linier adalah jenis potensiometer dimana *wiper* bergerak pada sepanjang jalur linier. Potensio linier juga dikenal sebagai *slider*, *pot slide*, atau *fader*, seperti pada gambar 2.18.



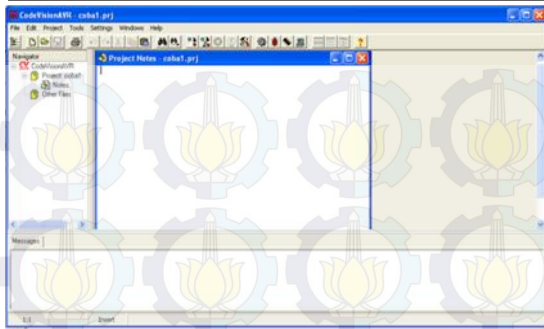
Gambar 2.18 Potensiometer Linier [8]

2. Potensiometer digital adalah potensiometer yang dikontrol secara elektronik. Dalam kebanyakan kasus mereka ada dari berbagai komponen resistif kecil secara seri. Setiap elemen resistif dilengkapi dengan saklar yang dapat berfungsi sebagai *tap-off point* atau posisi *wiper* sebenarnya.

2.6 CodeVisionAVR

Penggunaan mikrokontroler sekarang ini telah umum. Mulai dari penggunaan untuk kontrol sederhana sampai kontrol yang cukup kompleks, mikrokontroler dapat berfungsi jika telah diisi sebuah program, pengisian program ini dapat dilakukan menggunakan compiler yang selanjutnya di-download ke dalam mikrokontroler menggunakan downloader. Salah satu compiler program yang umum digunakan sekarang ini adalah CodeVision AVR [9] yang menggunakan bahasa pemrograman C [10].

CodeVision AVR mempunyai suatu keunggulan dari compiler lain, yaitu adanya *codewizard*, fasilitas ini memudahkan kita dalam inisialisasi mikrokontroler yang akan kita gunakan, codevision telah menyediakan konfigurasi yang bisa diatur pada masing-masing chip mikrokontroler yang akan kita gunakan, sehingga kita tidak perlu melihat *datasheet* untuk sekedar mengkonfigurasi mikrokontroler. Antarmuka CodeVision AVR dapat dilihat pada Gambar 2.19 Tampilan umum *Software* CodeVisionAVR.

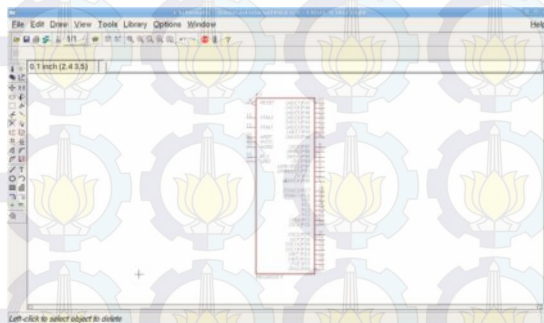


Gambar 2.19 Tampilan umum *Software CodeVisionAVR*.

2.7 Software EAGLE

Easily Applicable Graphical Layout Editor (EAGLE) merupakan sebuah aplikasi untuk mendesain skematik Elektronika maupun *Printed Circuit Board* (PCB). Dengan aplikasi EAGLE ini Kita bisa merancang, memodifikasi, dan mencetaknya untuk kemudian disablon ke dalam bentuk PCB.

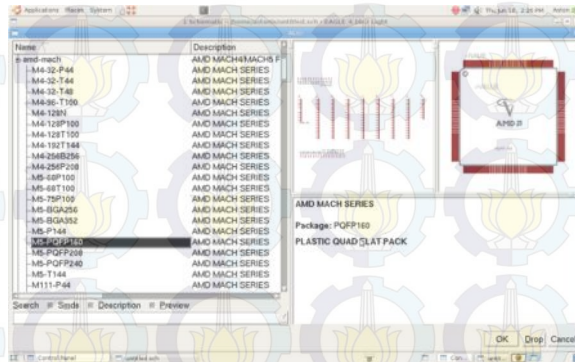
Schematic editor berfungsi untuk menggambar dan mengedit rangkaian skematik elektronika. Di sini Anda bisa menggunakan komponen-komponen yang ada dalam *Library* “Add”. Dapat dilihat pada Gambar 2.20 *Schematic Editor*.



Gambar 2.20 *Schematic Editor*

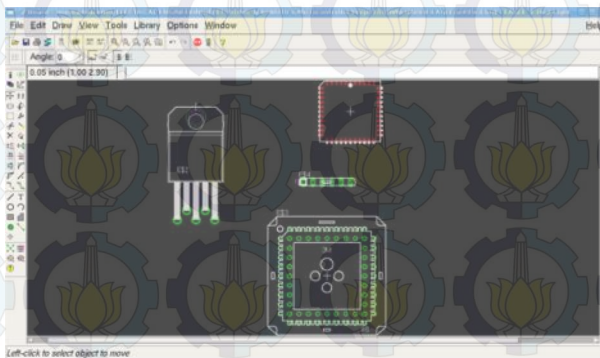


LibraryTool berfungsi untuk menambahkan berbagai komponen elektronika ke dalam *schematic editor*, tersedia bermacam-macam komponen mulai dari Komponen Diskret (Resistor, kapasitor, dioda) sampai IC Mikrokontroler dan Mikroprosesor. Dapat dilihat pada Gambar 2.21 *Library Tool*.



Gambar 2.21 *Library Tool*

Eagle Board berfungsi untuk mentransfer rangkaian skematik ke rangkaian PCB. Disini segala komponen bias ditata sesuai dengan yang diinginkan. Selain itu bias juga untuk me"routing" atau membuat jalur pada PCB. Dapat dilihat pada Gambar 2.22 *EAGLE Board*.



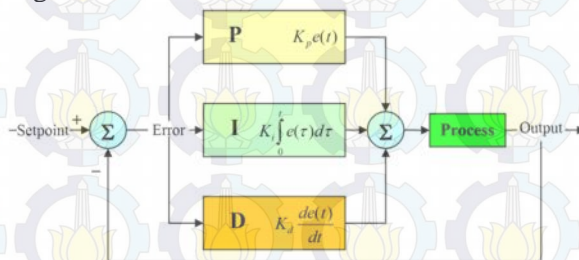
Gambar 2.22 *EAGLE Board*



2.8 Sistem PID

Instrumentasi dan control industri tentu tidak lepas dari sistem instrumentasi sebagai pengontrol yang digunakan dalam keperluan pabrik. Sistem kontrol pada pabrik tidak lagi manual seperti dahulu, tetapi saat sekarang ini telah dibantu dengan perangkat kontroler sehingga dalam proses produksinya suatu pabrik bisa lebih efisien dan efektif. Kontroler juga berfungsi untuk memastikan bahwa setiap proses produksi terjadi dengan baik.

PID merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut.[14] Pengontrol PID adalah pengontrol konvensional yang banyak dipakai dalam dunia industri. PID Blok Diagram dapat dilihat pada gambar 2.23 :



Gambar 2.23 PID Block Sistem

Adapun persamaan Pengontrol PID adalah :

$$mv(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (2.1)$$

Keterangan :

$mv(t)$ = output dari pengontrol PID atau *Manipulated Variable*

K_p = konstanta Proporsional



T_i = konstanta Integral

T_d = konstanta Derivatif

$e(t)$ = *error* (selisih antara *set point* dengan level aktual)

Persamaan Pengontrol PID diatas dapat juga dituliskan sebagai berikut :

$$mv(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (2.2)$$

Dengan :

$$K_i = K_p \times \frac{1}{T_i} \text{ dan } K_d = K_p \times T_d \quad (2.3)$$

Untuk lebih memaksimalkan kerja pengontrol diperlukan nilai batas minimum dan maksimum yang akan membatasi nilai *Manipulated Variable* yang dihasilkan.

Komponen kontrol PID ini terdiri dari tiga jenis yaitu Proportional, Integratif dan Derivatif. Ketiganya dapat dipakai bersamaan maupun sendiri-sendiri tergantung dari respon yang kita inginkan terhadap suatu *plant*.

2.8.1 Kontrol Proporsional

Kontrol P jika $G(s) = k_p$, dengan k adalah konstanta.

Jika $u = G(s) \cdot e$ maka $u = K_p \cdot e$ dengan K_p adalah Konstanta Proporsional. K_p berlaku sebagai *Gain* (penguat) saja tanpa memberikan efek dinamik kepada kinerja kontroler. Penggunaan kontrol P memiliki berbagai keterbatasan karena sifat kontrol yang tidak dinamik ini. Walaupun demikian dalam aplikasi-aplikasi dasar yang sederhana kontrol P ini cukup mampu untuk memperbaiki respon transien khususnya *rise time* dan *settling time*. Pengontrol proporsional memiliki keluaran yang sebanding/proporsional dengan

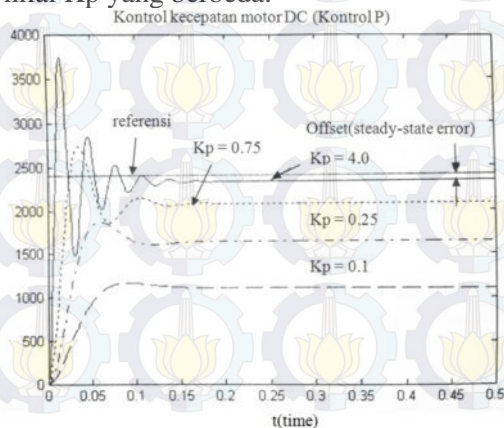


besarnya sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya).

Ciri-ciri pengontrol proporsional :

1. Jika nilai K_p kecil, pengontrol proporsional hanya mampu melakukan koreksi kesalahan yang kecil, sehingga akan menghasilkan respon sistem yang lambat (menambah *rise time*).
2. Jika nilai K_p dinaikan, respon/tanggapan sistem akan semakin cepat mencapai keadaan mantapnya (mengurangi *rise time*).
3. Namun jika nilai K_p diperbesar sehingga mencapai harga yang berlebihan, akan mengakibatkan sistem bekerja tidak stabil atau respon sistem akan berosilasi.
4. Nilai K_p dapat diset sedemikian sehingga mengurangi *steady state error*, tetapi tidak menghilangkannya.

Dalam uji simulasi ini, seluruh parameter motor dimasukkan ke dalam blok sistem simulasi. Kecepatan putar referensi yang digunakan adalah 2400 rpm. Skema kontrol P di atas diuji dengan memberikan nilai K_p yang berbeda.



Gambar 2.24 Respon P [14]



Nampak dalam gambar 2.24 bahwa semakin kecil K_p maka *offset* atau *steady state error* semakin besar. Namun nilai K_p yang terlalu besar akan menyebabkan osilasi pada saat start. Kontrol P dapat digunakan sendirian dalam aplikasi. Beberapa aplikasi seperti kontrol temperatur pada *heater*, sistem penghematan energi pada sistem *air conditioning* berdasarkan kontrol kecepatan motor kompresor sudah cukup memadai menggunakan kontrol P saja.

2.8.2 Kontrol Integratif

Pengontrol Integral berfungsi menghasilkan respon sistem yang memiliki kesalahan keadaan mantap nol (*Error Steady State* = 0). Jika sebuah pengontrol tidak memiliki unsur integrator, pengontrol proporsional tidak mampu menjamin keluaran sistem dengan kesalahan keadaan mantapnya nol.

Jika $G(s)$ adalah kontrol I maka u dapat dinyatakan sebagai $u(t) = [\text{integral } e(t)dt]K_i$, dengan K_i adalah konstanta Integral, dan dari persamaan di atas, $G(s)$ dapat dinyatakan sebagai $u = K_d \times [\Delta e / \Delta t]$. Jika $e(T)$ mendekati konstan (bukan nol) maka $u(t)$ akan menjadi sangat besar sehingga diharapkan dapat memperbaiki *error*. Jika $e(T)$ mendekati nol maka efek kontrol I ini semakin kecil. Kontrol I dapat memperbaiki sekaligus menghilangkan respon *steady-state*, namun pemilihan K_i yang tidak tepat dapat menyebabkan respon transien yang tinggi sehingga dapat menyebabkan ketidakstabilan sistem. Pemilihan K_i yang sangat tinggi justru dapat menyebabkan output berosilasi karena menambah orde sistem.

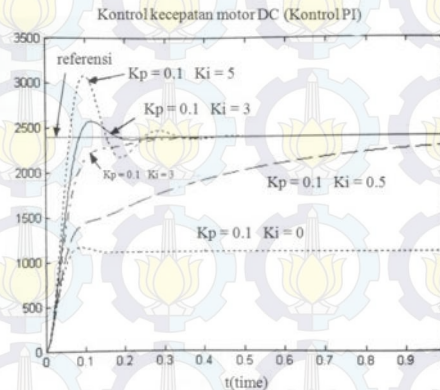
Keluaran pengontrol ini merupakan hasil penjumlahan yang terus menerus dari perubahan masukannya. Jika sinyal kesalahan tidak mengalami perubahan, maka keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Sinyal keluaran pengontrol integral merupakan luas bidang yang dibentuk oleh kurva kesalahan (*error*).

Diagram simulasi kontroler PI ini diuji pada 2400 rpm dengan K_p di set pada 0.1. Untuk melihat efek penerapan kontrol integral,



dibandingkan tiga kondisi hasil pemilihan parameter K_i yaitu 0, 0.5, dan 2. $K_i = 0$ menunjukkan bahwa kontroler adalah P saja.

Dalam Gambar 2.25 nampak bahwa penerapan kontrol P menurunkan steady-state erornya. Namun pemilihan K_i yang terlalu besar dapat menyebabkan sistem berosilasi saat *start*. Kontrol PI diketahui dipakai secara meluas di dunia industri. Dalam kebanyakan aplikasi di lapangan, kontrol PI dengan parameter yang di *tune* dengan baik dapat menyelesaikan berbagai permasalahan kontrol dengan cukup memadai dan relatif murah (tidak perlu kontrol komputer). Utamanya untuk kasus yang memiliki respon sistem relatif lambat, seperti kontrol temperatur, tekanan fluida, hidrolik dan lain sebagainya.



Gambar 2.25 Respon kontroler PI [14]

Ciri-ciri pengontrol integral :

1. Keluaran pengontrol integral membutuhkan selang waktu tertentu, sehingga pengontrol integral cenderung memperlambat respon.
2. Ketika sinyal kesalahan berharga nol, keluaran pengontrol akan bertahan pada nilai sebelumnya.
3. Jika sinyal kesalahan tidak berharga nol, keluaran akan menunjukkan kenaikan atau penurunan yang dipengaruhi oleh besarnya sinyal kesalahan dan nilai K_i .



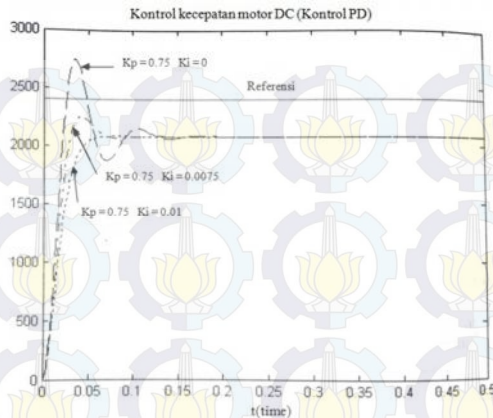
4. Konstanta integral K_i yang berharga besar akan mempercepat hilangnya *offset*. Tetapi semakin besar nilai konstanta K_i akan mengakibatkan peningkatan osilasi dari sinyal keluaran pengontrol.

2.8.3 Kontrol Derivatif

Keluaran pengontrol diferensial memiliki sifat seperti halnya suatu operasi derivatif. Perubahan yang mendadak pada masukan pengontrol akan mengakibatkan perubahan yang sangat besar dan cepat. Ketika masukannya tidak mengalami perubahan, keluaran pengontrol juga tidak mengalami perubahan, sedangkan apabila sinyal masukan berubah mendadak dan menaik (berbentuk fungsi *step*), keluaran menghasilkan sinyal berbentuk impuls. Jika sinyal masukan berubah naik secara perlahan (fungsi *ramp*), keluarannya justru merupakan fungsi *step* yang besar magnitudenya sangat dipengaruhi oleh kecepatan naik dari fungsi *ramp* dan faktor konstanta K_d .

Sinyal kontrol u yang dihasilkan oleh kontrol D dapat dinyatakan sebagai $G(s)=s.K_d$. Dari persamaan di atas, nampak bahwa sifat dari kontrol D ini dalam konteks “kecepatan” atau *rate* dari *error*. Dengan sifat ini ia dapat digunakan untuk memperbaiki respon transien dengan memprediksi *error* yang akan terjadi. Kontrol Derivatif hanya berubah saat ada perubahan *error* sehingga saat *error* statis kontrol ini tidak akan bereaksi, hal ini pula yang menyebabkan kontroler Derivatif tidak dapat dipakai sendiri.

Diagram simulasi kontroler PD ini diuji pada 2400 rpm dengan K_p di set pada 0.75, sedang K_d di set pada 0, 0.0075 dan 0.01. Untuk melihat efek penerapan simulasinya ditunjukkan dalam gambar 2.26.



Gambar 2.26 Respon kontroler PD [14]

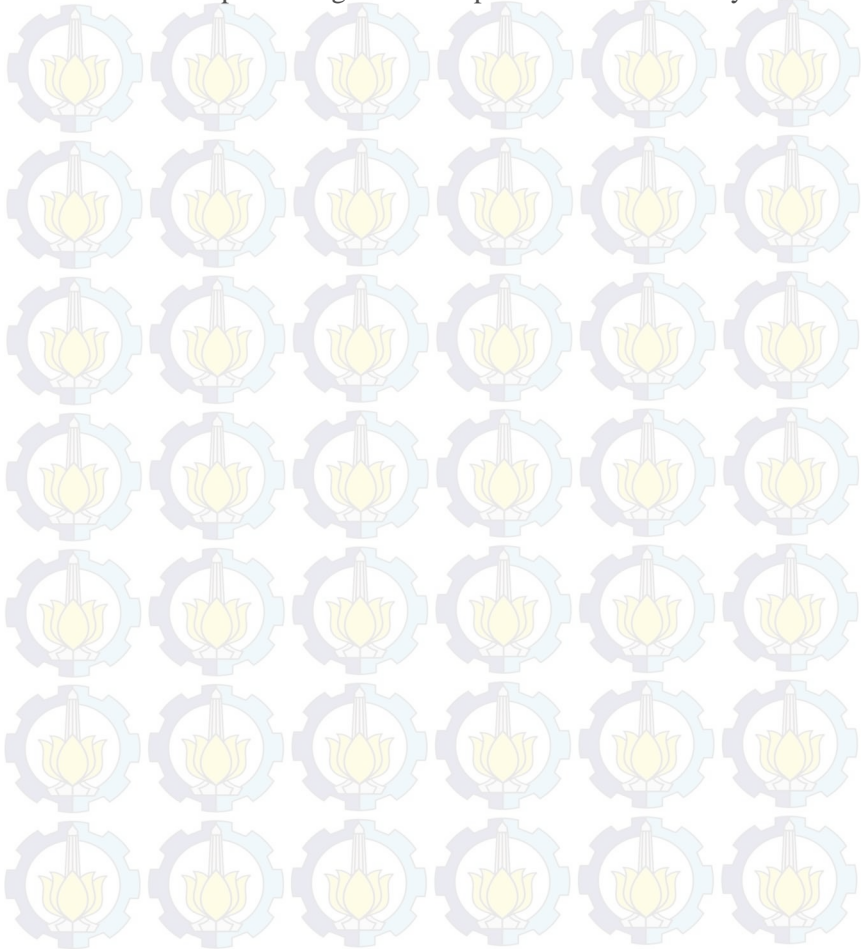
Dalam Gambar 2.26 nampak bahwa penerapan kontrol D dapat memperbaiki efek overshoot pada respon output. Namun pemilihan K_d yang terlalu besar dapat menyebabkan output tidak stabil dan dapat terjadi osilasi yang makin lama makin membesar.

Ciri-ciri pengontrol derivatif :

1. Pengontrol tidak dapat menghasilkan keluaran jika tidak ada perubahan pada masukannya (berupa perubahan sinyal kesalahan).
2. Jika sinyal kesalahan berubah terhadap waktu, maka keluaran yang dihasilkan pengontrol tergantung pada nilai K_d dan laju perubahan sinyal kesalahan.
3. Pengontrol diferensial mempunyai suatu karakter untuk mendahului, sehingga pengontrol ini dapat menghasilkan koreksi yang signifikan sebelum pembangkit kesalahan menjadi sangat besar. Jadi pengontrol diferensial dapat mengantisipasi pembangkit kesalahan, memberikan aksi yang bersifat korektif dan cenderung meningkatkan stabilitas sistem.
4. Dengan meningkatkan nilai K_d , dapat meningkatkan stabilitas sistem dan mengurangi *overshoot*.



Berdasarkan karakteristik pengontrol ini, pengontrol diferensial umumnya dipakai untuk mempercepat respon awal suatu sistem, tetapi tidak memperkecil kesalahan pada keadaan tunaknya. Kerja pengontrol diferensial hanyalah efektif pada lingkup yang sempit, yaitu pada periode peralihan. Oleh sebab itu pengontrol diferensial tidak pernah digunakan tanpa ada kontroler lainnya.

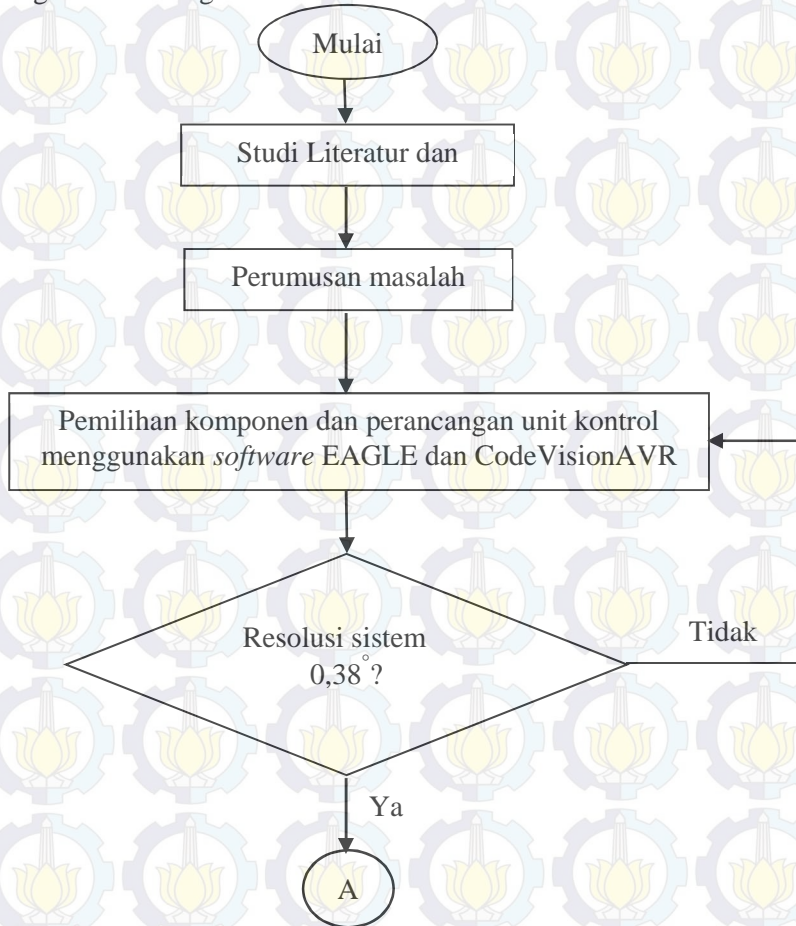




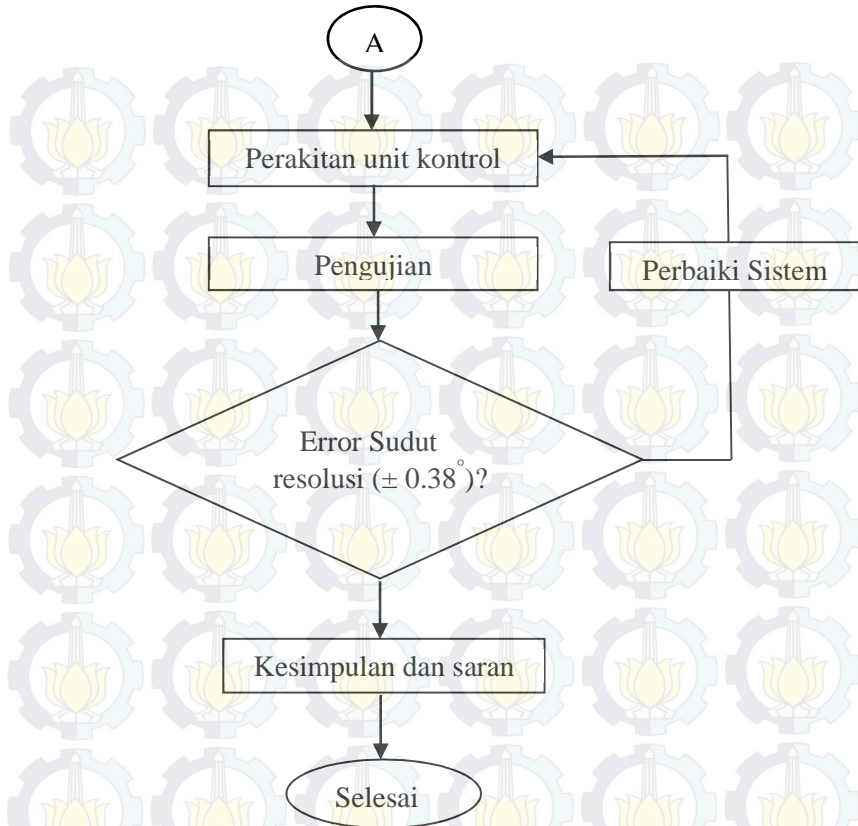
BAB III METODE PENELITIAN

3.1 Diagram Alir Penelitian

Penelitian pada tugas akhir ini dilaksanakan mengikuti diagram alir sebagai berikut:



Gambar 3.1 Diagram Alir Penelitian



Gambar 3.1 Diagram Alir Penelitian (Selesai)

3.2 Langkah-langkah penelitian

Berdasarkan diagram alir tersebut, langkah-langkah penelitian secara detail dijabarkan sebagai berikut:

1. Studi literatur dan Lapangan

Studi literatur dilakukan untuk memperoleh materi yang berkaitan dengan sistem kontrol pelontar peluru dengan dua sumbu putar. Literatur yang dikaji berasal dari jurnal dan buku.



2. Perumusan masalah

Langkah ini dilakukan untuk menentukan permasalahan yang akan dikaji dan dicari solusi terbaiknya. Dalam Tugas Akhir ini rumusan masalahnya adalah bagaimana merancang perangkat lunak sistem kontrol yang akan diaplikasikan pada pelontar peluru dengan dua sumbu putar.

3. Perancangan dan pemilihan komponen

Pada tahap ini dilakukan perancangan sistem kontrol. Perancangan tersebut pemilihan komponen yang akan digunakan pada sistem. Pemilihan ini disesuaikan dengan kebutuhan sistem yang akan dibuat. Contohnya sensor memiliki spesifikasi sesuai target pengukuran, jenis motor memiliki torsi yang cukup untuk menggerakkan beban, mikrokontroler memiliki spesifikasi sesuai kebutuhan dan sebagainya.

4. Simulasi rancangan menggunakan *software*

Pada tahap ini dilakukan perakitan komponen menjadi perangkat keras atau *hardware* sesuai rancangan pada *software* EAGLE. Selanjutnya dibuat program menggunakan *software* CodeVisionAVR yang akan digunakan pada mikrokontroler agar dapat menjalankan fungsinya. Perangkat keras ini kemudian diuji agar dapat berfungsi dengan baik.

5. Resolusi $< \pm 0,38^\circ$.

Setelah perancangan berupa desain perangkat keras dengan komponen yang telah dipilih selesai, perangkat keras diatur agar resolusi kecermatan sistem tidak melebihi $\pm 0,38^\circ$.

6. Pengujian

Langkah ini dilakukan untuk mengetahui bahwa pembuatan sistem kontrol telah memenuhi target.



Tugas Akhir Manufaktur

7. Kepresisian = resolusi

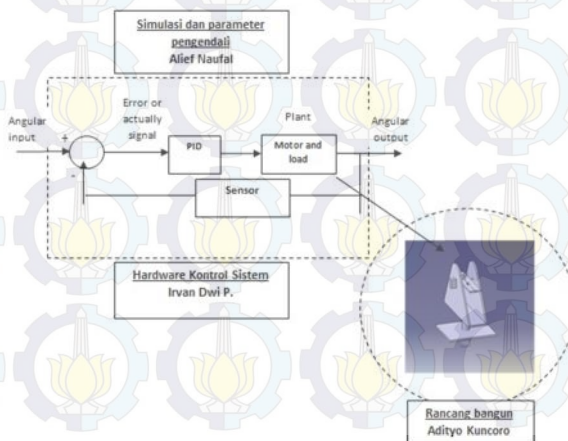
Sistem kontrol memenuhi target jika hasil kecermatan pembacaan alat ukur sama dengan $\pm 0.38^\circ$.

8. Kesimpulan dan saran

Kesimpulan dari penelitian Tugas Akhir dapat diambil jika sistem kontrol telah bekerja dengan baik. Kesimpulan berupa nilai error yang didapat dari pengujian sistem kontrol pelontar peluru dengan dua sumbu putar. Saran berupa masukan untuk pengembangan sistem kontrol pelontar peluru dengan dua sumbu putar pada khususnya dan perangkat keras unit kontrol lain pada umumnya.

3.3 Blok Diagram Sistem

Sistem kontrol yang dipakai pada tugas akhir ini menggunakan sistem kontrol PID. PID merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut. Pengontrol PID adalah pengontrol konvensional yang banyak dipakai dalam dunia industri. Blok diagram berfungsi memetakan proses pada sistem dari *input* hingga menjadi *output* yang diharapkan. Blok diagram sistem yang akan digunakan ditunjukkan pada Gambar 3.2.



Gambar 3.2 Blok Diagram Sistem



Blok diagram pada sistem kontrol pelontar peluru tersebut dapat dijelaskan sebagai berikut:

- *Input* pada sistem berupa sudut inklinasi dan rotasi yang diharapkan. Batas jarak inklinasi $+60^\circ$ dan jarak rotasi $\pm 60^\circ$. *Input* dimasukkan ke dalam sistem kontrol melalui perangkat lunak atau *software*.
- Minimum sistem mikrokontroler merupakan unit yang dapat diprogram dan menjalankan fungsinya sesuai program tersebut. Mikrokontroler yang digunakan pada sistem kontrol ini adalah IC ATmega32.
- Motor sebagai aktuator yang berfungsi mengubah energi listrik menjadi besaran mekanik. Pada sistem ini, aktuator berupa motor listrik yang menghasilkan gerak rotasi. Motor yang digunakan berjenis motor servo.

Feedback yang digunakan pada tugas akhir ini adalah sensor sudut yang berfungsi sebagai *error detector*. Jika sudut tersebut telah sesuai dengan data input, maka minimum sistem akan mengirim sinyal berhenti ke motor. Pembaca sudut yang digunakan adalah potensiometer yang terdapat di dalam motor servo. *Output* yang dihasilkan dari sistem kontrol adalah sudut pelontar peluru dan target sesuai dengan input posisi pada perangkat lunak. Sudut ini diharapkan memiliki kecermatan $\pm 0.38^\circ$.



[Halaman Ini Sengaja Dikosongkan]



BAB IV

PERANCANGAN DAN PERAKITAN

4.1 Spesifikasi Komponen

Proses pembuatan perangkat keras (*hardware*) unit kontrol untuk alat Pelontar Peluru Dengan Dua Sumbu Putar ini memerlukan beberapa macam komponen. Komponen yang digunakan memiliki spesifikasi seperti pada tabel berikut ini.

1. Mikrokontroler

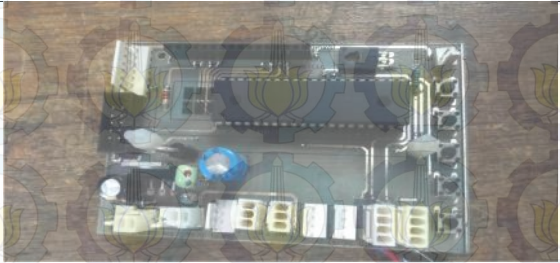
Tabel 4.1 Mikrokontroler ATMega32

Fungsi	Sebagai pusat kontrol yang dapat memberi, menerima dan mengolah data atau sinyal dari komponen lain.
Merk	Atmel
Type	ATMega32A
Gambar	
Alasan pemilihan komponen	<ul style="list-style-type: none">- ATmega 32 memiliki 4 Port sehingga memungkinkan digunakan pada unit kontrol yang membutuhkan 3 Port (sensor, motor, serial)- ATmega 32 memiliki ADC sebesar 10 bit sehingga cukup baik digunakan dalam pengukuran- Harga lebih murah dibandingkan ATmega lain dengan spesifikasi sejenis



2. Sistem Minimum

Tabel 4.2 Sistem Minimum

Fungsi	Merupakan rangkaian yang menghubungkan mikrokontroler dengan komponen lain sehingga dapat menjalankan fungsinya.
Merk	-
Tipe	-
Gambar	
Alasan pemilihan komponen	<ul style="list-style-type: none"> - Sistem sesuai dengan yang dibutuhkan. - Pengaturan program sesuai dengan yang diinginkan.

3. USB to TTL Serial

Tabel 4.3 USB to TTL Serial

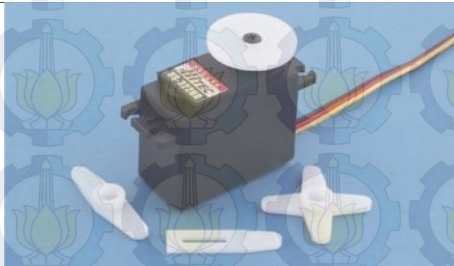
Fungsi	Menghubungkan minimum sistem dan port USB pada computer.
Merk	D-SUN
Gambar	



Alasan pemilihan komponen	Konektor serial dengan port USB dipilih dikarenakan lebih praktis dalam penggunaannya dibandingkan konektor RS 232. Pada beberapa jenis komputer port USB terletak di bagian depan sedangkan port RS 232 terletak di bagian belakang. Pada beberapa jenis komputer jinjing (<i>laptop</i>) tidak terdapat port RS 232 dan terdapat port USB.
---------------------------	--

4. Motor Servo

Tabel 4.4 Motor Servo

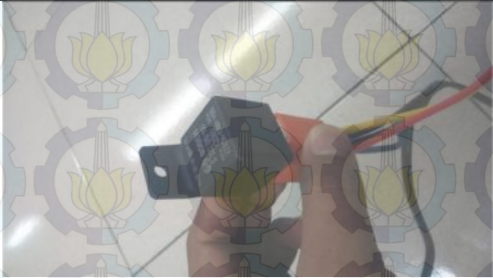
Fungsi	Sebagai komponen aktuator yang mengubah sinyal data dari driver menjadi gerak mekanis berupa putaran
Merk	Hitec HS 322HD
Gambar	
Alasan pemilihan komponen	<ul style="list-style-type: none">- Memiliki gerakan per step sehingga lebih presisi.- Dapat memberikan putaran dan torsi penuh.- Torsi maksimum mendekati nilai yang diinginkan.

5. Relay 5 pin

Tabel 4.5 Relay 5 pin

Fungsi	komponen untuk mengalirkan listrik dengan
--------	---



	arus yang besar dengan yang menggunakan kendali listrik dengan arus kecil.
Merk	Bosch
Tipe	250 12V 30A
Gambar	
Alasan pemilihan komponen	<ul style="list-style-type: none"> - komponen untuk mengalirkan listrik dengan arus yang besar dengan yang menggunakan kendali listrik dengan arus kecil. - untuk melindungi Motor ataupun komponen lainnya dari kelebihan Tegangan ataupun hubung singkat (<i>Short</i>).

6. Pembaca Sudut

Tabel 4.6 Potensiometer

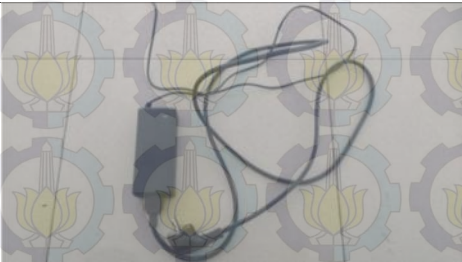
Fungsi	Potensiometer yang dioperasikan oleh suatu mekanisme dapat digunakan sebagai transduser.
Jenis Pembacaan	Digital
Merk	Alpha



Gambar	
Alasan pemilihan komponen	<ul style="list-style-type: none">- Memiliki tingkat kepresisian pengukuran yang baik.- Tidak ada <i>noise</i>- Mudah digunakan.

7. Adapter listrik 2.5A


Tabel 4.7 Adapter Listrik 2.5A

Fungsi	Mengatur besar tegangan dan arus listrik sesuai kebutuhan. Adapter ini digunakan sebagai sumber tegangan relay.
Merk	Kemafur
Tipe	220/250V
Gambar	
Alasan pemilihan komponen	Adapter dengan arus keluaran 2.5 A digunakan sebagai sumber tegangan relay yang digunakan. <i>Output</i> dari adapter berupa DC 12 V, 2.5 A sesuai dengan input relay.



8. Baterai

Tabel 4.8 Baterai

Fungsi	Alat listrik-kimiawi yang menyimpan energi dan mengeluarkannya dalam bentuk listrik. Digunakan untuk memberikan <i>supply</i> daya untuk mikrokontroler.
Merek	Lipo
Tipe	20c 2200mAh 12V.
Gambar	
Alasan pemilihan komponen	<ul style="list-style-type: none"> - Sebagai penyuplai daya <i>portable</i>. - Memiliki kapasitas penyimpanan daya yang sangat besar.

9. Downloader USBasp

Tabel 4.9 Downloader USBasp

Fungsi	- Jembatan penghubung antara sistem minimum dengan PC agar program dapat diisikan dari PC ke dalam mikrokontroler Atmega pada sistem minimum.
Merek	Baite
Tipe	USBASP Programmer (Double Power 3.3V/5V)



Gambar	
Alasan pemilihan komponen	- Sebagai downloader program.

4.2 Perakitan (*Assembly*) Perangkat Keras

Motor servo diletakkan pada dudukan yang telah dibuat. Dudukan ini dihubungkan dengan kaki meja alat pelontar peluru menggunakan rumah motor yang di las. Pada gambar 4.1 dan gambar 4.2 terlihat poros motor telah terhubung dengan poros pelontar peluru menggunakan *fleksibel joint*. Jika diberi tegangan dan sinyal dari driver maka motor servo berputar dan menggerakkan poros.

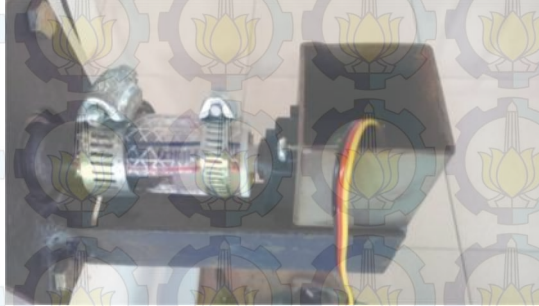


Gambar 4.1 Susunan Motor Servo arah Rotasi, Rumah Dudukan, Dan *Fleksibel Joint*

Motor servo ini memiliki spesifikasi 0.19 detik per 60⁰ pada saat tidak ada beban. Torsi maksimumnya sebesar 3kg.cm pada saat diberikan daya 4.8V. Sudut maksimum yang bisa ditempuh



sebesar 180^0 . Motor servo ini digunakan untuk bergerak rotasi $\pm 60^0$.



Gambar 4.2 Susunan Motor Servo arah Inklinasi, Rumah Dudukan, Dan *Fleksibel Joint*

Motor servo ini memiliki spesifikasi 0.19 detik per 60^0 pada saat tidak ada beban. Torsi maksimumnya sebesar 3kg.cm pada saat diberikan daya 4.8V. Sudut maksimum yang bisa ditempuh sebesar 180^0 . Motor servo ini digunakan untuk bergerak inklinasi $+60^0$. Potensiometer terletak pada bagian dalam motor servo. Fungsi potensiometer pada alat pelontar peluru adalah sebagai hambatan dengan nilai ADC maksimal 1024. pada sistem ini nilai potensiometer ditentukan dengan melakukan kalibrasi antara sudut dan nilai pada potensiometer. Pada potensiometer motor servo arah rotasi, nilai potensiometer didapat sebesar 790 pada sudut -60^0 , 880 pada sudut 0^0 dan 970 pada sudut $+60^0$. Sedangkan pada potensiometer motor servo arah inklinasi nilai potensiometer didapat 900 pada sudut 0^0 dan 960 pada sudut 60^0 . Dengan menggunakan rumus :

$$x = m.z + c \quad (4.1)$$

Untuk potensiometer pada motor servo arah rotasi.

$$y = m.z + c \quad (4.2)$$

Untuk potensiometer pada motor servo arah inklinasi.

Dimana :

x = nilai potensiometer pada arah rotasi



y = nilai potensiometer pada arah inklinasi

m = nilai gradient

c = nilai konstanta

z = sudut

Contoh perhitungan menentukan nilai potensiometer pada motor servo arah rotasi sudut -60° , sudut 0° dan sudut $+60^\circ$ dengan menggunakan rumus 4.1.

Menentukan nilai c pada sudut 0°

$$x = m \cdot z + c$$

$$880 = m(0^\circ) + c$$

$$880 = c, \text{ nilai } c \text{ adalah } 880$$

Menentukan nilai m pada sudut $+60^\circ$

$$x = m \cdot z + c$$

$$970 = m(+60^\circ) + 880$$

$$970 - 880 = m(+60^\circ)$$

$$90 = m(+60^\circ)$$

$$m = 1 \frac{1}{2}$$

sehingga didapatkan :

$$x = 1 \frac{1}{2} \cdot z + 880$$

Contoh perhitungan menentukan nilai potensiometer pada motor servo arah inklinasi sudut -60° dengan menggunakan rumus 4.2.

Menentukan nilai c pada sudut 0°

$$y = m \cdot z + c$$



$$900 = m(0^\circ) + c$$

$$900 = c, \text{ nilai } c \text{ adalah } 900$$

Menentukan nilai m pada sudut 60°

$$y = m.z + c$$

$$960 = m(60^\circ) + 900$$

$$960 - 900 = m(60^\circ)$$

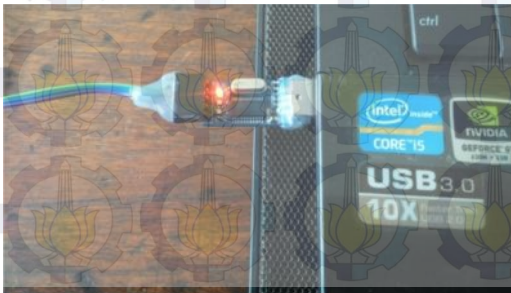
$$60 = m(60^\circ)$$

$$m = 1$$

sehingga didapatkan :

$$y = 1.z + 900$$

Konektor USB to TTL Serial berfungsi menghubungkan mikrokontroler dan komputer. Sambungan dengan mikrokontroler menggunakan kabel melalui PortD sistem minimum terlihat pada gambar 4.3. Komponen USB to TTL Serial langsung dihubungkan dengan port USB pada komputer. Indikator LED warna merah menunjukkan terdapat proses transfer data yang sedang terjadi, baik dari mikrokontroler menuju komputer atau sebaliknya.



Gambar 4.3 USB To TTL Serial Terhubung Dengan Komputer



Unit kontrol yang telah tersusun sesuai skema dapat dilihat pada Gambar 4.4.



Gambar 4.4 Susunan *Hardware* Pelontar Peluru

Dari gambar tersebut dapat dilihat bahwa letak dan sambungan tiap komponen sudah sesuai ketentuan fungsi masing-masing. Penyusunan kabel penghubung juga diatur agar tidak menghambat gerak pelontar peluru. Dengan demikian perangkat keras (*hardware*) pelontar peluru dapat berfungsi dengan baik.

4.3 Resolusi Sistem

4.3.1 Resolusi motor servo

Untuk mendapatkan resolusi motor servo, diperlukan beberapa pengaturan. Pengaturan tersebut dilakukan sedemikian



rupa sehingga mempunyai frekuensi 50Hz dengan nilai TOP pada timer = 27647 / 0x6BFF.

Dengan spesifikasi motor servo, didapat sudut maksimum motor berputar sebesar 180°. Untuk mengetahui resolusi motor servo, sudut maksimal dibagi dengan nilai ADC pada ATmega32,

$$\begin{aligned}\text{Resolusi Motor Servo} &= \left(\frac{180^\circ}{1024} \right) \\ &= 0.17578^\circ\end{aligned}$$

Hal tersebut menunjukkan bahwa perubahan nilai ADC sebesar 1 bit menyebabkan perubahan sudut sebesar 0.17578°.

Dengan analogi penskalaan, maka didapatkan besar nilai:

- OCR 862 untuk menghasilkan sinyal 1ms, maka Servo mengatur posisi untuk sudut -90°.
- OCR 1978 untuk menghasilkan sinyal 1.5ms, maka Servo mengatur posisi untuk sudut 0°.
- OCR 3197 untuk menghasilkan sinyal 2ms, maka Servo mengatur posisi untuk sudut 90°.
- 1219 adalah nilai selisih antara sudut 90° dengan 0°.
- 1115 adalah nilai selisih antara sudut -90° dengan 0°.

4.3.2 Resolusi Potensiometer

Pada sistem kontrol pelontar peluru diperlukan sebuah sensor pembaca untuk membuktikan apakah sistem sesuai dengan parameter yang diinginkan. Potensiometer diset pada setiap motor servo.

- Potensiometer pada motor servo arah rotasi Kalibrasi nilai potensiometer didapat sebesar 790 pada sudut -60, 880 pada sudut 0° dan 970 pada sudut +60°. Jumlah nilai ADC dari -60° hingga +60° sebesar 180 bit. Range sudut antara -60° hingga +60° sebesar 120°. Sehingga dapat diketahui resolusi sudut dengan rumus :

$$\frac{120^\circ}{180} = 0.667^\circ$$



Hal tersebut menunjukkan bahwa perubahan nilai ADC sebesar 1 menyebabkan perubahan sudut sebesar 0.667° .

- Potensiometer pada motor servo arah inklinasi Kalibrasi nilai potensiometer didapat sebesar 900 pada sudut 0° dan 960 pada sudut 60° . Jumlah nilai ADC dari 0^0 hingga 60° sebesar 60. Range sudut antara -60° hingga $+60^\circ$ sebesar 60° . Sehingga dapat diketahui resolusi sudut potensiometer pada pelontar peluru adalah:

$$\frac{60^\circ}{60} = 1^\circ$$

Hal tersebut menunjukkan bahwa perubahan nilai ADC sebesar 1 bit menyebabkan perubahan sudut sebesar 1° .

4.3.3 Kepresisian Sistem Kontrol

ATMega32 memiliki nilai ADC 10 bit atau sebesar 2^{10} satuan data. *Range* tegangan yang masuk ke sistem ADC dapat diketahui dengan perumusan sebagai berikut.

$$b_1 2^{-1} + b_2 2^{-2} + \dots + b_n 2^{-n} = \frac{V_{in}}{V_R}$$

dimana b_1, b_2, \dots, b_n = n-bit keluaran digital
 V_{in} = tegangan masuk analog
 V_R = tegangan referensi analog.

Tegangan referensi analog adalah tegangan input yang digunakan untuk pengoperasian ADC. Tegangan ini berupa V_{cc} sebesar 5 V. Tegangan input memiliki *range* antara tegangan minimum dan tegangan maksimum. Tegangan input minimum bernilai 0 V sehingga nilai ADC yang dihasilkan adalah 0. Tegangan input maksimum dapat diketahui dari perhitungan berikut.

$$b_1 2^{-1} + b_2 2^{-2} + \dots + b_n 2^{-n} = \frac{V_{in}}{V_R}$$
$$2^{-1} + 2^{-2} + \dots + 2^{-10} = \frac{V_{in}}{5}$$



Tugas Akhir Manufaktur

$$V_{in} = 5 \times 0,999023438$$

$$V_{in} = 4,99 \text{ Volt}$$

Dari hasil perhitungan tersebut, *range* tegangan masuk sistem ADC sebesar 0 hingga 4,99511719 V. Nilai ADC maksimal dapat diperoleh dalam bentuk desimal dengan perhitungan sebagai berikut.

$$N = \text{INT} \left(\frac{V_{in}}{V_R} 2^n \right)$$

$$N = \text{INT} \left(\frac{4,99}{5} 2^{10} \right)$$

$$N = \text{INT} (1023,9)$$

$$N = 1023_{10}$$

ADC memiliki nilai maksimum sebesar 1023 dalam bilangan desimal. Nilai ini diperoleh ketika tegangan yang masuk dalam sistem ADC mencapai nilai maksimum, yaitu sebesar 4,99 volt.

Perhitungan resolusi ADC dapat diketahui dari rumus berikut, dimana tegangan referensi adalah nilai tegangan input ($V_{cc} = 5 \text{ V}$).

$$\Delta V = V_R \times 2^{-n}$$

$$\Delta V = 5 \times 2^{-10}$$

$$\Delta V = 0,004883 \text{ volt}$$

$$\Delta V = 4,883 \text{ milivolt}$$

Perubahan nilai ADC yang dihasilkan oleh perubahan tegangan input sebesar 4,883 mV dapat diketahui dengan perumusan sebagai berikut.



$$N = INT \left(\frac{V_{in}}{V_R} 2^n \right)$$

$$N = INT \left(\frac{0,004883}{5} 2^{10} \right)$$

$$N = INT (1,0000384)$$

$$N = 1$$

Hal tersebut menunjukkan bahwa perubahan tegangan input sebesar 4,883 mV menyebabkan perubahan nilai ADC sebesar 1 bit.

4.4 Algoritma Sistem

Pada proses pembuatan perangkat lunak diperlukan suatu perencanaan yang berupa algoritma-algoritma pengolahan citra supaya tujuan akhir dari perangkat lunak ini tercapai. Algoritma umum dari perangkat lunak ini yaitu:

1. Sistem pada kondisi diam dan semua perangkat keras telah tersambung dengan mikrokontroler.
2. Tekan tombol *switch on/off* untuk menyalakan mikrokontroler. Pada saat ini kompresor dalam keadaan menyala, dan solenoid valve dalam keadaan mati.
3. Untuk menginputkan data dari komputer terlebih dahulu dilakukan penekanan pada push button S2. Pada penekanan tersebut *push button* mengirimkan kode yang memungkinkan menginput data via komputer.
4. Selanjutnya adalah menginputkan data berupa sudut. Data tersebut masuk ke mikrokontroler kemudian diteruskan ke motor servo. Motor servo bergerak ke arah koordinat yang diinputkan, jika sudut sesuai dengan pembacaan busur derajat maka input sudut berhasil.

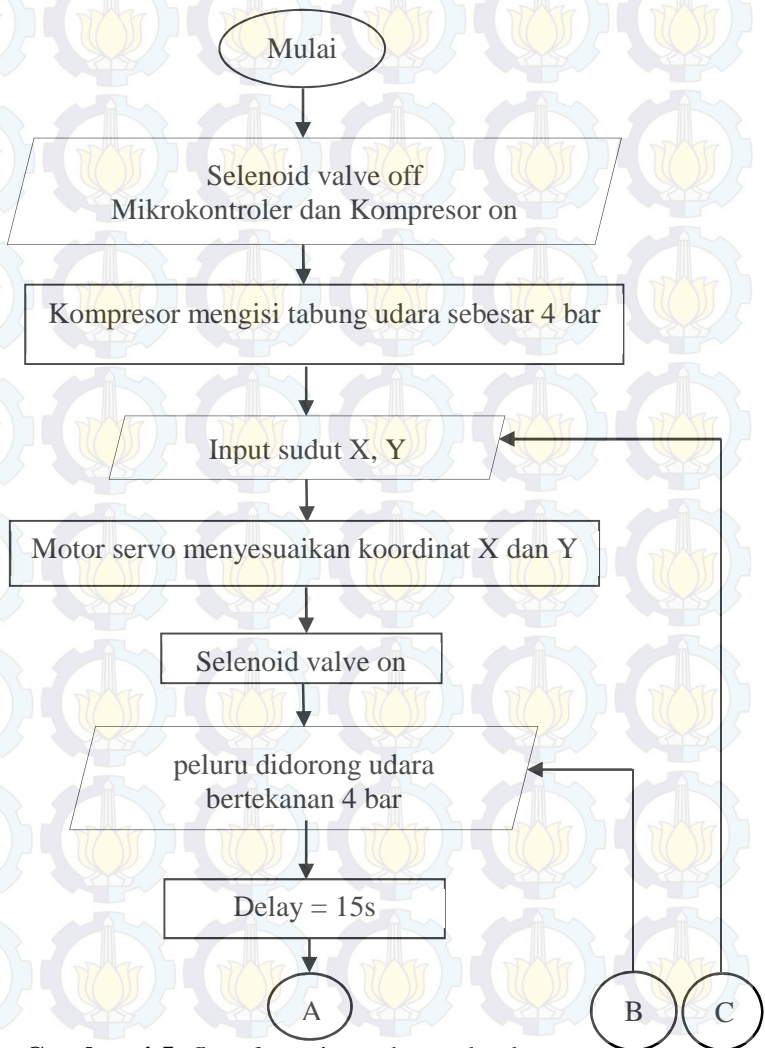


5. Setelah motor servo bergerak menuju target, perintah selanjutnya dari program adalah menyalakan *solenoid valve* yang akan meneruskan tekanan dari kompresor ke laras pelontar peluru untuk mendorong peluru plastik keluar dari laras.
6. Solenoid valve hanya menyala untuk beberapa saat, setelah itu mati kembali. Setelah proses pelontar pertama selesai, kompresor menyala kembali untuk mengisi tekanan yang hilang.
7. Mengulang proses 5 dan 6 untuk pelontaran peluru hingga 5 kali pelontaran.
8. Selanjutnya kembali mengulang proses 4-7 dengan input sudut yang berbeda. Percobaan dilakukan sebanyak empat kali pada tiga titik yang ditentukan.

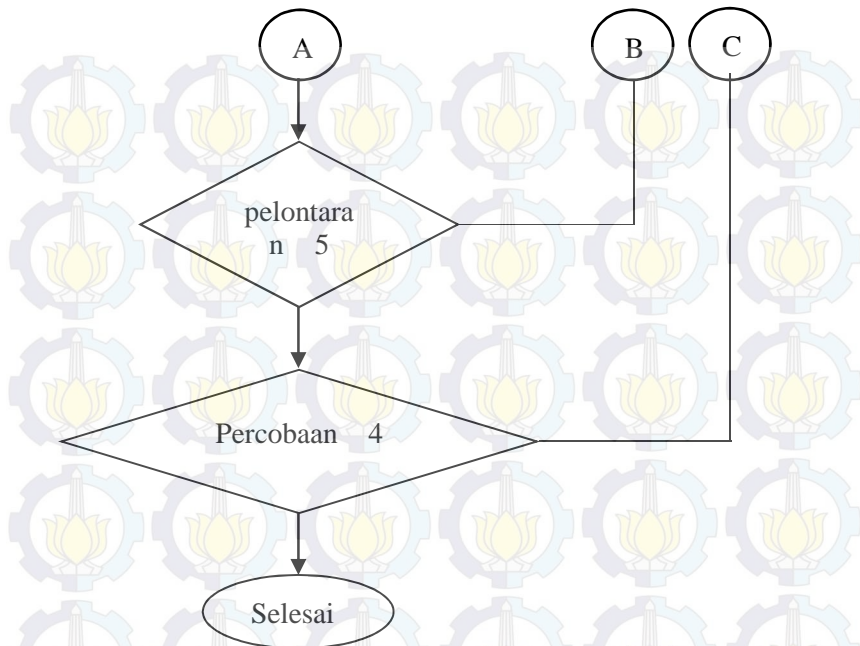


4.4.1 Algoritma Sistem Kontrol Pelontar Peluru

Gambar 4.5 adalah flowchart untuk sistem control pelontar peluru



Gambar 4.5 flowchart sistem kontrol pelontar peluru



Gambar 4.5 flowchart sistem kontrol pelontar peluru (selesai)

4.5 Perancangan *Interface* pada Microsoft Visual Basic 6.0

Microsoft Visual Basic 6.0 digunakan untuk membuat *Interface* program pada komputer. Desain *Interface* program sesuai dengan yang diinginkan. Gambar 4.6 adalah tampilan *Interface* pada Visual Basic 6.0.



Form1

Nilai X: Send

Nilai Y: Exit

COM Port: SET

>> 2 Digit Input Data untuk Nilai X dan Y

Ex : X= +23 dan Y=20
X= -23 dan Y=20

Gambar 4.6 Interface pada Visual Basic 6.0

Pada program tersebut terdapat 3 kolom, dimana kolom paling atas berupa input untuk nilai sumbu X, kolom tengah berupa input nilai sumbu Y dan kolom dibawah berupa posisi kabel USB to TTL yang dihubungkan dengan port USB di computer. Di sisi kanan terdapat tombol Send untuk mengirim data ke mikrokontroler dan tombol Exit untuk menutup jendela interface program.



[Halaman Ini Sengaja Dikosongkan]



BAB V

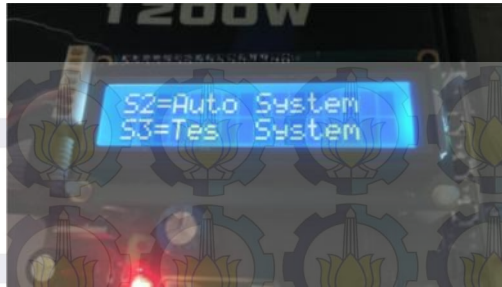
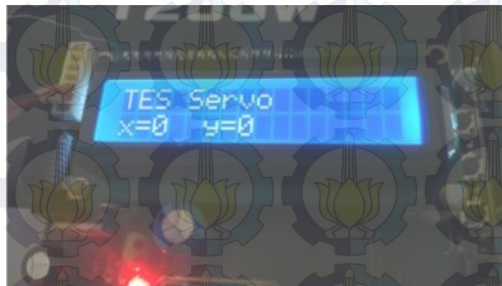
PENGUJIAN DAN ANALISA

5.1 Pengujian

Pada pengujian ini adalah menghubungkan semua komponen dengan minimum system dari mikrokontroler. Dimana prosedurnya adalah sebagai berikut :

1. Nyalakan minimum sistem yang juga terhubung dengan power supply 12V untuk *supply* mikrokontroler.
2. Hubungkan modul I/O yang berupa LCD dan *push button* ke minimum sistem. LCD dihubungkan ke PORTB dan *push button* ke PORTA.
3. Pastikan *power supply* 12V sudah terhubung dengan baik, lalu hubungkan dengan mikrokontroler.
4. Hubungkan motor servo arah inklinasi dan arah rotasi dengan mikrokontroler.
5. Hubungan kabel USB to TTL dari mikrokontroler ke PC yang berguna untuk meng-*input* data.
6. Hubungkan *relay* dengan mikrokontroler.
7. Hubungkan kedua potensiometer dengan mikrokontroler.
8. Untuk mendownload program, hubungkan kabel downloader dari mikrokontroler dengan PC.
9. Tekan tombol *on/off* untuk menyalakan mikrokontroler.

Data yang diterima oleh mikrokontroler tersebut diinput menggunakan program yang digabungkan dengan *interface* dari *Visual Basic*. Gambar 5.1 menunjuk bahwa mikrokontroler siap digunakan sehingga pada tampilan LCD bertuliskan “S2 = *Auto System*” dan “S3 = *Test System*”. Untuk “S2 = *Auto System*” layar LCD tidak menampilkan keterangan dikarenakan program berjalan pada komputer. Untuk tes sudut secara manual, tombol *push button* S3 yang ditekan. Gambar 5.2 menunjukkan pada saat tes sudut manual.

**Gambar 5.1** Tampilan LCD**Gambar 5.2** Tes *Input* Manual Sudut

Pada saat menekan tombol *push button* S2, mode penginput-an menjadi berubah. Untuk menginput data sudut, jalankan program *interface* seperti pada gambar 5.3.

Form1

Nilai X:

Nilai Y:

COM Port:

Send

Exit

SET

>> 2 Digit Input Data untuk Nilai X dan Y

Ex: X= +23 dan Y=20
X= -23 dan Y=20

Gambar 5.3 Interface Program Pelontar Peluru



5.2 Pengujian Perangkat keras (*Hardware*)

Hardware pada system ini terdiri dari *Power Supply*, Minimum Sistem, *Push button*, *Relay*, LCD, dan juga motor servo. Akan dilakukan pengujian sebanyak empat kali pada tiga koordinat sasaran tembak yang ditentukan. Untuk pengujian sudut yang ditentukan sebesar $(-60^\circ, 30^\circ)$, pengujian selanjutnya sebesar $(0^\circ, 30^\circ)$ dan setelah itu pengujian terakhir sebesar $(+60^\circ, 30^\circ)$. Pengujian tersebut dilakukan sebanyak empat kali.

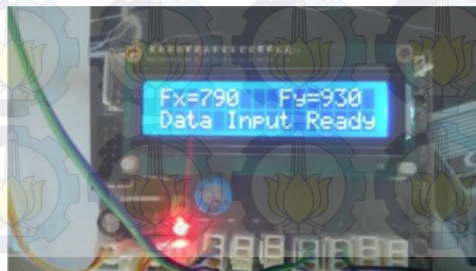
- Pengujian pertama titik $(-60^\circ, 30^\circ)$, $(0^\circ, 30^\circ)$ dan $(+60^\circ, 30^\circ)$.

Gambar 5.4 hingga Gambar 5.5 adalah pengujian pertama sistem kontrol dalam menggerakkan motor servo rotasi dan motor servo inklinasi.



Gambar 5.4 Pengujian Sudut $(-60^\circ, 30^\circ)$

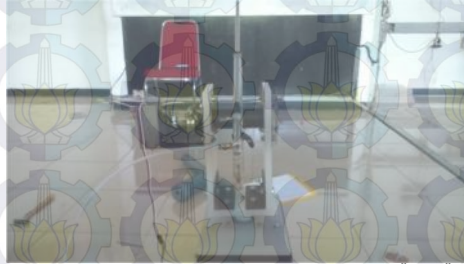
Pada pengujian titik $(-60^\circ, 30^\circ)$ gambar 5.4 motor servo sumbu rotasi dan motor servo sumbu inklinasi bergerak sesuai dengan data penginputan dari *interface software* pada laptop.



Gambar 5.5 Nilai ADC dari percobaan sudut $(-60^\circ, 30^\circ)$



Setelah proses penginputan data dilakukan dan motor servo bergerak sesuai data penginputan, didapat nilai pembacaan ADC dari potensiometer pada arah rotasi sebesar 790 dan arah inklinasi sebesar 930 seperti pada Gambar 5.5.



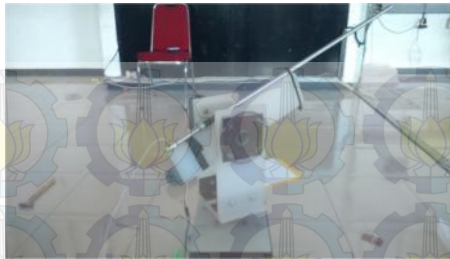
Gambar 5.6 Pengujian Sudut ($0^{\circ}, 30^{\circ}$)

Pada pengujian titik ($0^{\circ}, 30^{\circ}$) Gambar 5.6 motor servo sumbu rotasi dan motor servo sumbu inklinasi bergerak sesuai dengan data penginputan dari *interface software* pada laptop.



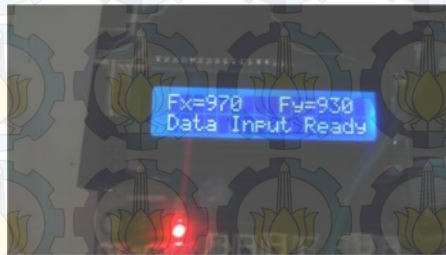
Gambar 5.7 Nilai ADC dari percobaan sudut ($0^{\circ}, 30^{\circ}$)

Setelah proses penginputan data dilakukan dan motor servo bergerak sesuai data penginputan, didapat nilai pembacaan ADC dari potensiometer pada arah rotasi sebesar 880 dan arah inklinasi sebesar 930 seperti pada gambar 5.7.



Gambar 5.8 Pengujian Sudut ($+60^\circ, 30^\circ$)

Pada pengujian titik ($+60^\circ, 30^\circ$) Gambar 5.8 motor servo sumbu rotasi dan motor servo sumbu inklinasi bergerak sesuai dengan data penginputan dari *interface software* pada laptop.



Gambar 5.9 Nilai ADC dari percobaan sudut ($+60^\circ, 30^\circ$)

Setelah proses peng-input-an data dilakukan dan motor servo bergerak sesuai data penginputan, didapat nilai pembacaan ADC dari potensiometer pada arah rotasi sebesar 970 dan arah inklinasi sebesar 930 seperti pada Gambar 5.9.

Setelah melakukan empat kali percobaan pada ketiga titik, didapat nilai-nilai ADM yang beragam, seperti pada Table 5.1.



Tabel 5.1 Hasil Percobaan

Percobaan Ke-	(-60°,30°)		(0°,30°)		(60°,30°)	
1	790	930	881	930	971	930
2	789	931	880	931	972	929
3	790	932	880	930	970	931
4	792	930	882	930	970	930

5.3 Analisa Hasil Pengujian

5.3.1 Analisa Hasil Pengujian Percobaan Pada Titik (-60°,30°)

5.3.1.1 Perhitungan nilai rata-rata

Dari Pengujian Percobaan Pada Titik (0°,30°) sebanyak empat kali didapat nilai ADC hasil pembacaan potensiometer pada tabel 5.2.

Tabel 5.2 Tabel perhitungan nilai rata-rata

Percobaan Ke-	X(-60°)	Y(30°)
1	790	930
2	789	931
3	790	932
4	792	930

Perhitungan rata-rata (\bar{x})

$$\bar{x} = \sum \frac{xi}{n} = \frac{x1 + x2 + x3 + x4}{n} = \frac{790 + 789 + 790 + 792}{4} = 790.25$$

Perhitungan rata-rata (\bar{y})

$$\bar{y} = \sum \frac{yi}{n} = \frac{y1 + y2 + y3 + y4}{n} = \frac{930 + 931 + 932 + 930}{4} = 930.85$$

Dari perhitungan nilai rata-rata, didapatkan nilai \bar{x} sebesar 790.25 dan nilai \bar{y} sebesar 930.85. Dari nilai tersebut



dikonversikan menjadi sudut dengan persamaan rumus 4.3 untuk nilai x dan persamaan rumus 4.4 nilai \bar{y} .

Untuk nilai x ,

$$x = 1 \frac{1}{2} \cdot z + 880$$

maka,

$$\bar{x} = 1 \frac{1}{2} \cdot z + 880$$

$$z = \frac{\bar{x} - 880}{1 \frac{1}{2}}$$

$$z = \frac{790.25 - 880}{1 \frac{1}{2}}$$

$$z = -59.833^\circ$$

untuk nilai y ,

$$y = 1 \cdot z + 900$$

$$\bar{y} = 1 \cdot z + 900$$

$$z = \frac{\bar{y} - 900}{1}$$

$$z = \frac{930.58 - 900}{1}$$

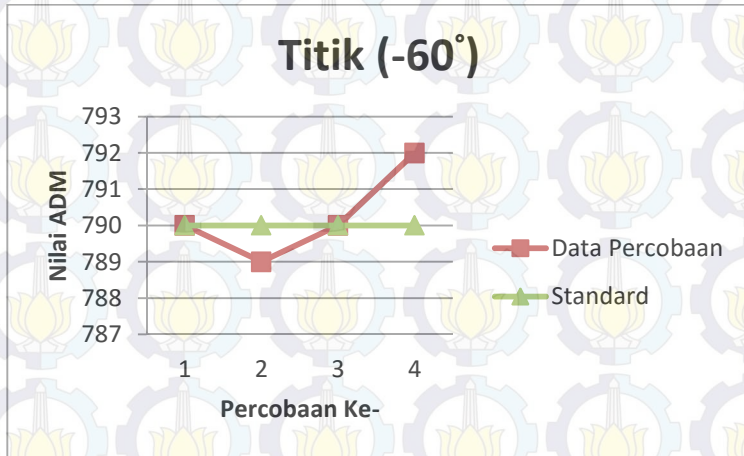
$$z = 30.58^\circ$$

Dari hasil perhitungan didapat nilai z pada sumbu x sebesar -59.833° . Nilai tersebut dibandingkan dengan nilai acuan $-60^\circ + (\pm 0.38^\circ)$. Nilai z tersebut masih berada didalam *range* $\pm 0.38^\circ$ yaitu -59.62° .

Dari hasil perhitungan didapat nilai z pada sumbu y sebesar 30.58° . Nilai tersebut dibandingkan dengan nilai acuan $30^\circ + (\pm 0.38^\circ)$. Nilai z tersebut berada diluar *range* $\pm 0.38^\circ$ yaitu 30.38° .

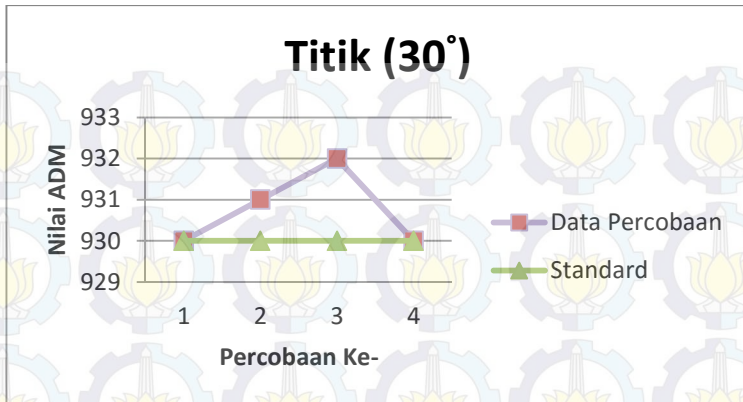


5.3.1.2 Pembahasan Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik $(-60^{\circ}, 30^{\circ})$.



Gambar 5.10 Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik (-60°)

Dari Gambar 5.10 grafik perbandingan data percobaan dengan data acuan terlihat bahwa di antara keempat percobaan ada hasil percobaan yang datanya sama dengan data acuan yaitu 790. Range hasil percobaan adalah yang terbesar 792 dan yang terkecil adalah 789. Dari pembahasan grafik perbandingan data percobaan dengan data acuan dan hasil perhitungan nilai rata-rata, didapatkan data hasil percobaan sesuai dengan data acuan.



Gambar 5.11 Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik (30°)

Dari Gambar 5.11 grafik perbandingan data percobaan dengan data acuan terlihat bahwa di antara keempat percobaan ada hasil percobaan yang datanya sama dengan data acuan yaitu 930. Range hasil percobaan adalah yang terbesar 932 dan yang terkecil adalah 930. Dari pembahasan grafik perbandingan data percobaan dengan data acuan dan hasil perhitungan nilai rata-rata, didapatkan data hasil percobaan tidak sesuai dengan data acuan.

5.3.2 Analisa Hasil Pengujian Percobaan Pada Titik (0° , 30°)

5.3.2.1 Perhitungan nilai rata-rata

Dari Pengujian Percobaan Pada Titik (0° , 30°) sebanyak empat kali didapat nilai ADC hasil pembacaan potensiometer pada tabel 5.3.

Tabel 5.3 Tabel perhitungan nilai rata-rata

Percobaan Ke-	X(0°)	Y(30°)
1	881	930
2	880	931
3	880	930
4	882	930



Perhitungan rata-rata (\bar{x})

$$\bar{x} = \sum \frac{xi}{n} = \frac{x1 + x2 + x3 + x4}{n} = \frac{881 + 880 + 880 + 882}{4} = 880.75$$

Perhitungan rata-rata (\bar{y})

$$\bar{y} = \sum \frac{yi}{n} = \frac{y1 + y2 + y3 + y4}{n} = \frac{930 + 931 + 930 + 930}{4} = 930.25$$

Dari perhitungan nilai rata-rata, didapatkan nilai \bar{x} sebesar 880.75 dan nilai \bar{y} sebesar 930.25. Dari nilai tersebut dikonversikan menjadi sudut dengan persamaan rumus 4.3 untuk nilai \bar{x} dan persamaan rumus 4.4 nilai \bar{y} .

Untuk nilai \bar{x} ,

$$x = 1 \frac{1}{2} \cdot z + 880$$

maka,

$$\bar{x} = 1 \frac{1}{2} \cdot z + 880$$

$$z = \frac{x - 880}{1 \frac{1}{2}}$$

$$z = \frac{880.75 - 880}{1 \frac{1}{2}}$$

$$z = 0.5$$

untuk nilai \bar{y} ,

$$y = 1 \cdot z + 900$$

$$\bar{y} = 1 \cdot z + 900$$

$$z = \frac{\bar{y} - 900}{1}$$

$$z = \frac{930.25 - 900}{1}$$

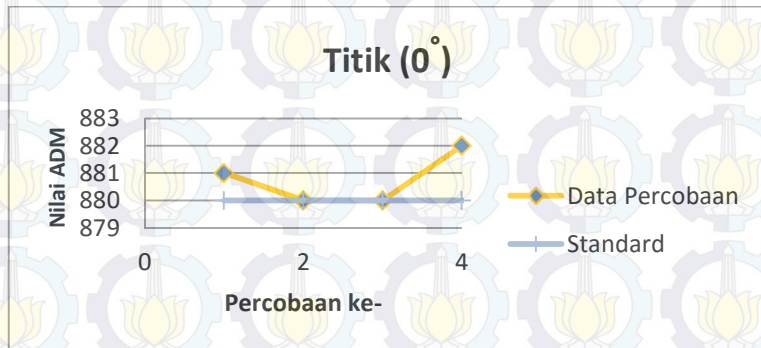
$$z = 30.25^\circ$$



Dari hasil perhitungan didapat nilai z pada sumbu x sebesar 0.5° . Nilai tersebut dibandingkan dengan nilai acuan $-60^\circ + (\pm 0.38^\circ)$. Nilai z tersebut berada diluar *range* $\pm 0.38^\circ$ yaitu 0.38° .

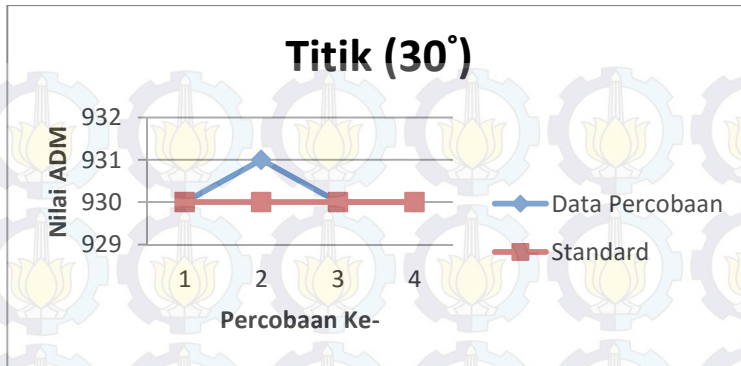
Dari hasil perhitungan didapat nilai z pada sumbu y sebesar 30.25° . Nilai tersebut dibandingkan dengan nilai acuan $30^\circ + (\pm 0.38^\circ)$. Nilai z tersebut berada didalam *range* $\pm 0.38^\circ$ yaitu 30.38° .

5.3.2.2 Pembahasan Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik ($0^\circ, 30^\circ$).



Gambar 5.12 Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik (0°)

Dari Gambar 5.12 grafik perbandingan data percobaan dengan data acuan terlihat bahwa di antara keempat percobaan ada hasil percobaan yang datanya sama dengan data acuan yaitu 880. Range hasil percobaan adalah yang terbesar 882 dan yang terkecil adalah 880. Dari pembahasan grafik perbandingan data percobaan dengan data acuan dan hasil perhitungan nilai rata-rata, didapatkan data hasil percobaan tidak sesuai dengan data acuan.



Gambar 5.13 Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik (30°)

Dari Gambar 5.13 grafik perbandingan data percobaan dengan data acuan terlihat bahwa di antara keempat percobaan ada hasil percobaan yang datanya sama dengan data acuan yaitu 930. *Range* hasil percobaan adalah yang terbesar 931 dan yang terkecil adalah 930. Dari pembahasan grafik perbandingan data percobaan dengan data acuan dan hasil perhitungan nilai rata-rata, didapatkan data hasil percobaan sesuai dengan data acuan.

5.3.3 Analisa Hasil Pengujian Percobaan Pada Titik ($+60^{\circ}, 30^{\circ}$)

5.3.3.1 Perhitungan nilai rata-rata

Dari Pengujian Percobaan Pada Titik ($0^{\circ}, 30^{\circ}$) sebanyak empat kali didapat nilai ADC hasil pembacaan potensiometer pada tabel 5.4.

Tabel 5.4 Tabel perhitungan nilai rata-rata

Percobaan Ke-	X($+60^{\circ}$)	Y(30°)
1	971	930
2	972	929
3	970	931
4	970	930



Perhitungan rata-rata (\bar{x})

$$\bar{x} = \sum \frac{xi}{n} = \frac{x1 + x2 + x3 + x4}{n} = \frac{971 + 972 + 970 + 970}{4} = 970.75$$

Perhitungan rata-rata (\bar{y})

$$\bar{y} = \sum \frac{yi}{n} = \frac{y1 + y2 + y3 + y4}{n} = \frac{930 + 929 + 931 + 930}{4} = 930$$

Dari perhitungan nilai rata-rata, didapatkan nilai x sebesar 970.75 dan nilai \bar{y} sebesar 930. Dari nilai tersebut dikonversikan menjadi sudut dengan persamaan rumus 4.3 untuk nilai x dan persamaan rumus 4.4 nilai \bar{y} .

Untuk nilai x ,

$$x = 1 \frac{1}{2} \cdot z + 880$$

maka,

$$\bar{x} = 1 \frac{1}{2} \cdot z + 880$$

$$z = \frac{x - 880}{1 \frac{1}{2}}$$

$$z = \frac{970.75 - 880}{1 \frac{1}{2}}$$

$$z = 60.5$$

untuk nilai y ,

$$y = 1 \cdot z + 900$$

$$\bar{y} = 1 \cdot z + 900$$

$$z = \frac{\bar{y} - 900}{1}$$

$$z = \frac{930 - 900}{1}$$

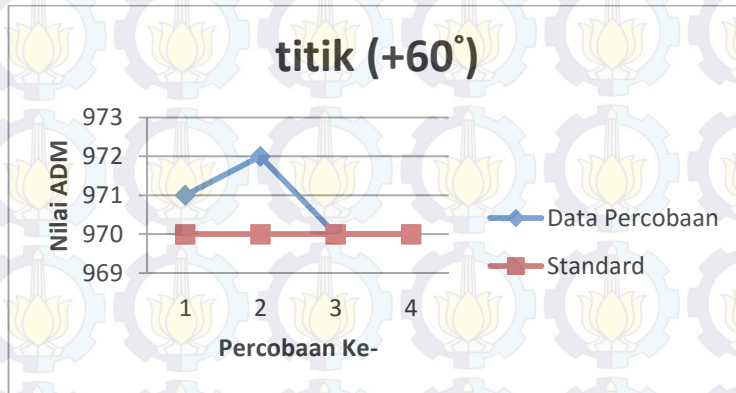
$$z = 30$$



Dari hasil perhitungan didapat nilai z pada sumbu x sebesar $+60.5^\circ$. Nilai tersebut dibandingkan dengan nilai acuan $-60^\circ + (\pm 0.38^\circ)$. Nilai z tersebut berada diluar $range \pm 0.38^\circ$ yaitu $+60.38^\circ$.

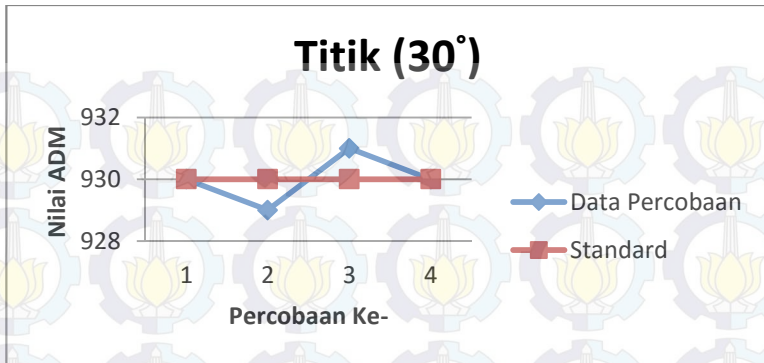
Dari hasil perhitungan didapat nilai z pada sumbu y sebesar 30° . Nilai tersebut dibandingkan dengan nilai acuan $30^\circ + (\pm 0.38^\circ)$. Nilai z tersebut berada didalam $range \pm 0.38^\circ$ yaitu 30.38° .

5.3.3.2 Pembahasan Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik $(+60^\circ, 30^\circ)$.



Gambar 5.14 Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik $(+60^\circ)$

Dari Gambar 5.14 grafik perbandingan data percobaan dengan data acuan terlihat bahwa di antara keempat percobaan ada hasil percobaan yang datanya sama dengan data acuan yaitu 970. Range hasil percobaan adalah yang terbesar 972 dan yang terkecil adalah 970. Dari pembahasan grafik perbandingan data percobaan dengan data acuan dan hasil perhitungan nilai rata-rata, didapatkan data hasil percobaan tidak sesuai dengan data acuan.



Gambar 5.15 Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik (30°)

Dari Gambar 5.15 grafik perbandingan data percobaan dengan data acuan terlihat bahwa di antara keempat percobaan ada hasil percobaan yang datanya sama dengan data acuan yaitu 930. Range hasil percobaan adalah yang terbesar 931 dan yang terkecil adalah 929. Dari pembahasan grafik perbandingan data percobaan dengan data acuan dan hasil perhitungan nilai rata-rata, didapatkan data hasil percobaan sesuai dengan data acuan.



[Halaman Ini Sengaja Dikosongkan]



BAB VI PENUTUP

6.1 Kesimpulan

Setelah melakukan pengujian dan analisa, maka dapat diambil beberapa kesimpulan tentang sistem kerja dari sistem yang telah dibuat, sebagai berikut :

1. Rancang bangun sistem kontrol pelontar peluru yang terdiri dari mikrokontroler, dua motor servo dengan nilai torsi maksimal sebesar 3.7 kg.cm, dua potensiometer, pengatur relay dan LCD pembaca nilai potensiometer. Komponen-komponen tersebut berasal dari merk yang beredar dipasaran, ditunjukkan pada Tabel 6.1.

Tabel 6.1 Komponen Sistem Kontrol Pelontar Peluru

No.	Komponen	Merk	Type	Jumlah
1	IC	Atmel	ATMega32A	1
2	Motor Servo	Hitec HS 322HD	Digital	2
3	Potensiometer	Alpha	Digital	2
4	Downloader	Baite	USBASP Programmer	1
5	ASB to TTL Serial	D-SUN	Konektor Serial	1
6	Baterai	Lipo	20c 2200mAh 12V	1

2. Didapat Resolusi Motor servo, resolusi potensiometer dan resolusi sistem yang telah dikalibrasikan dengan sistem kontrol. Untuk resolusi motor servo sebesar 0.17578° per 1 bit. Resolusi potensiometer sebesar 0.667° per 1 bit. Resolusi sistem sebesar 4.883 mV per 1 bit.
3. Dari ketiga pengujian yang telah dilakukan dapat dilihat pembacaan sudut dengan bantuan potensiometer mempunyai variasi nilai. Nilai-nilai tersebut ada yang



berada diantara *range* $\pm 0.38^\circ$ dan ada yang berada diluar *range* $\pm 0.38^\circ$.

6.2 Saran

Pada penelitian tugas akhir ini, masih terdapat beberapa kekurangan, maka beberapa saran yang diperlukan dalam proses perbaikan-perbaikan diantaranya adalah:

1. Masih terjadi banyak *noise* dalam percobaan tes sudut, sehingga nilai yang didapat berubah-ubah, oleh karena itu sebaiknya proses instalasi dilakukan secara hati-hati.
2. Pembuatan alat yang *portable*, sehingga alat tidak perlu melakukan proses pencarian pola melalui *personal computer*.
3. Perlu penataan kembali letak komponen-komponen perangkat kerasnya.

DAFTAR PUSTAKA

- [1] Nasyir, T., Pramujati, B., Nurhadi, H., dan Pitowarno, E., 2014. "Control Simulation of An Automatic Turret Gun Based on Force Control Method", IEEE Xplore Digital Library 1,1:13-18
- [2] Dong, G., dan Kuang, G. 2014. "*Target Recognition via Information Aggregation Through Dempster-Shafer's Evidence Theory*", IEEE Geosci, Remote Sens. Lett., 12,1:1247-1251
- [3] Chandra, Y., Purnomo, Didik S., dan Suryawati N., Endah. 2011. "Rancang Bangun Sistem *Ballast* Pada ROV (*Remotely Underwater Vehicle*)", Jurusan Teknik Mekatronika, Kampus PENS-ITS
- [4] Shue, S., Hargrove. C., dan Conrad, james., 2012. "Low Cost Semi-Autonomous Sentry Robot", IEEE Southeastcon Proceedings 1,1:1-5
- [5] Baizhan, Y., dan Yuan, Z. 2010. "*Modeling and Simulating for Bomber Air Turret System*", IEEE 1,1:1-4
- [6] Pramana, A., Mei. 2012. Belajar Mudah memprogram Mikrokontroler ATmega (Part I), <URL:<https://labsistemkontrol.wordpress.com/2012/05/03/belajar-mudah-memprogram-mikrokontroller-atmega-8535-part-i/>>.
- [7] Purnama, A., Mei. 2012. Motor Servo, <URL:<http://elektronika-dasar.web.id/teori-elektronika/motor-servo/>>
- [8] Piscataway. 2000. "*The Authoritative Dictionary of IEEE Standards Terms (IEEE 100)* (ed. edisi ketujuh)", IEEE Press, New Jersey, ISBN 0-7381-2601-2.
- [9] Andrianto, Heri, 2008, " Pemograman Mikrokontroler AVR ATMEGA 16 Menggunakan Bahasa C (Code Vision AVR)", Bandung, INFORMATIKA.

[10] Balafif, N., Nopember. 2013. Mengenal pemrograman C dan C++,

<URL:<http://staff.unipdu.ac.id/nufan/2013/11/11/mengenal-bahasa-pemrograman-c-dan-c/>>

[11] Eliezer, I., Oktober. 2014. Mengenal PWM, <URL:<http://www.geyosoft.com/2014/pwm-pulse-width-modulation>>.

[12] Andri, M. November. 2013. Pulse Width Modulation (PWM), <URL:http://andri_mz.staff.ipb.ac.id/pulse-width-modulation-pwm/>

[13] Nasution, F., 2011. “Perancangan Telemetry Suhu Ruang berbasis Mikrokontroler ATmega”, Universitas Sumatera Utara.

[14] Nise, N., 2011. “*Control Systems Engineering Sixth Edition*”, California State Polytechnic University, Pomona.

[15] Mandika, H., 2014. Istilah dan Pengertian Potensiometer, <URL:<http://www.academia.edu/5279049/Potensiometer>>

LAMPIRAN

Lampiran A Coding Sistem Kontrol Pelontar Peluru Dengan Dua Sumbu Putar

This program was produced by the
CodeWizardAVR V2.05.3 Standard
Automatic Program Generator

© Copyright 1998-2011 Pavel
Haiduc, HP
InfoTech s.r.l.
<http://www.hpinfotech.com>

Project :

Version :

Date : 24/06/2015

Author : Irvan Dwi Putra

Company : P3.Corp

Comments:

Chip type : ATmega32A

Program type : Application

AVR Core Clock frequency:
11,059200
MHz

Memory model : Small

External RAM size : 0

Data Stack size : 512

```
*****  
*****  
*****  
*/
```

```
#include <mega32a.h>
```

```
#include <stdio.h>
```

```
#include <alcd.h>
```

```
#include <delay.h>
```

```
#include <stdlib.h>
```

```
#define ADC_VREF_TYPE 0x00
```

```
#define FRAMING_ERROR (1<<FE)
```

```
#define PARITY_ERROR (1<<UPE)
```

```
#define DATA_OVERRUN  
(1<<DOR)
```

```
#define DATA_REGISTER_EMPTY  
(1<<UDRE)
```

```
#define RX_COMPLETE (1<<RXC)
```

```
#define Servo_Ver 0
```

```
#define Servo_Hor 1
```

```
#define PB1 PINC.7
```

```
#define PB2 PINC.6
```

```
#define PB3 PINC.5
```

```
#define PB4 PINC.4
```

```
#define BUZZER PORTD.6
```

```
#define SHOOT PORTD.3
```

```
#ifndef RXB8
```

```
#define RXB8 1
```

```
#endif
```

```
#ifndef TXB8
```

```
#define TXB8 0
```

```
#endif
```

```
#ifndef UPE
```

```
#define UPE 2
```

```
#endif
```

```
#ifndef DOR
```

```
#define DOR 3
```

```
#endif
```

```
#ifndef FE
```

```
#define FE 4
```

```
#endif
```

```
#ifndef UDRE
```

```
#define UDRE 5
```

```
#endif
```

```
#ifndef RXC
```

```
#define RXC 7
```

```
#endif
```

```
//Global Variable
```

```
int x,y;
```

```
char cetak[16];
```

```
unsigned int pot_x,pot_y;
```

```
int nilaix,nilaiy;
```

```
char
```

```
Proses_Parsing  
,Data_Terima[2  
,x_cons[1],X_V  
alue[5],Y_Valu  
e[5],Count_Co  
de,Buff_Code[  
10],i,other[100  
,Tes_Buff[10];
```

```
int Nilai_X,Nilai_Y;
```

```
int Error;
```

```
unsigned int
```

```
Konst_P,SP,PID  
_Kontrol;
```

```
// External Interrupt 0 service  
routine
```

```
interrupt [EXT_INT0] void  
ext_int0_isr(vo  
id)
```

```
{
```

```
// Place your code here
```

```
}
```

```
// External Interrupt 1 service  
routine
```

```
interrupt [EXT_INT1] void  
ext_int1_isr(vo  
id)
```

```
{
```

```
// Place your code here
```

```
}
```

```
// USART Receiver buffer
```

```
#define RX_BUFFER_SIZE 8
```

```
char rx_buffer[RX_BUFFER_SIZE];
```

```
#if RX_BUFFER_SIZE <= 256
```

```
unsigned char
```

```
rx_wr_index,rx  
_rd_index,rx_c  
ounter;
```

```
#else
```

```
unsigned int
```

```
rx_wr_index,rx  
_rd_index,rx_c  
ounter;
```

```
#endif
```

```
// This flag is set on USART  
Receiver buffer  
overflow
```

```
bit rx_buffer_overflow;
```

```
// USART Receiver interrupt  
service routine
```

```
interrupt [USART_RXC] void  
usart_rx_isr(vo  
id)
```

```
{
```

```
char status,data;  
status=UCSRA;
```

```
data=UDR;
```


/*

```
if(Proses_Parsing==1)
```

```
{
```

```
if(data=='*') {Count_Code=0;}
```

```
{
```

```
Buff_Code[Count_Code]=data;
```

```
Count_Code++;
```

```
//format kirim = $misi,code  
input, code  
output//  
*,1,0,0
```

```
if(Buff_Code[0]!='*')
```

```
{
```

```
nilaix=0;
```

```
while(data!=',')
```

```
{
```

```
X_Value[nilaix]=data;
```

```
nilaix++;
```

```
}
```

```
}
```

```
Count_Code=0;
```

```
}
```

```
}
```

*/

```
if(Proses_Parsing==1)
```

```
{
```

```
if(data=='*') {Count_Code=0;}
```

```
{
```

```
Buff_Code[Count_Code]=data;
```

```
Count_Code++;
```

```
//format kirim = $misi,code  
input, code  
output//  
*,1,0,0
```

```
if(Buff_Code[0]!='*')
```

```
{
```

```
for(i=0;i<=5;i++){
```

```
if(i<=2){X_Valu  
e[i]=Buff_Code  
[i+1];}
```

```
Y_Value[i]=Buff_Code[i+5];
```

```
//lcd_clear();
```

```
}
```

```
Nilai_X=atoi(X_Value);
```

```
if(Nilai_X>60)Nilai_X=60;
```

```
if(Nilai_X<-60)Nilai_X=-60;
```

```
Nilai_Y=atoi(Y_Value);
```

```
if(Nilai_Y>60)Nilai_Y=60;
```

```
//lcd_clear();
```

```
//lcd_gotoxy(0,0);
```

```
//sprintf(cetak,"Input  
Sudut");lcd_puts(cetak);
```

```
//lcd_gotoxy(0,0);
```

```
//lcd_putsf("Input  
SudutServo");
```

```
//lcd_gotoxy(0,0);
```

```
sprintf(cetak,"X=%3d  
Y=%2d",Nilai_X  
,Nilai_Y);lcd_puts(cetak);
```

```
Count_Code=0;
```

```
}
```

```
if ((status & (FRAMING_ERROR |  
PARITY_ERROR  
|  
DATA_OVERRUN))!=0)
```

```
{  
rx_buffer[rx_wr_index++]=data;
```

```
#if RX_BUFFER_SIZE == 256
```

```
// special case for receiver buffer  
size=256
```

```
if (++rx_counter == 0)  
rx_buffer_overflow=1;
```

```
#else
```

```
if (rx_wr_index ==  
RX_BUFFER_SIZE)  
rx_wr_index=0;
```

```
if (++rx_counter ==  
RX_BUFFER_SIZE)
```

```
{  
rx_counter=0;
```

```
rx_buffer_overflow=1;
```

```
}
```

```
#endif
```

```

    }

}

#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART
// Receiver buffer
#define _ALTERNATE_GETCHAR_

#pragma used+
char getchar(void)
{
    char data;
    while (rx_counter==0);
    data=rx_buffer[rx_rd_index++];
    #if RX_BUFFER_SIZE != 256
    if (rx_rd_index ==
        RX_BUFFER_SIZE)
        rx_rd_index=0;
    #endif
    #asm("cli")
    --rx_counter;
    #asm("sei")
    return data;
}

```

```

    }

#pragma used-
#endif

// Read the AD conversion result
unsigned int read_adc(unsigned
    char
    adc_input)
{
    ADMUX=adc_input |
        (ADC_VREF_TYPE
        PE & 0xff);

    // Delay needed for the
    // stabilization of
    // the ADC input
    // voltage
    delay_us(10);

    // Start the AD conversion
    ADCSRA |= 0x40;

    // Wait for the AD conversion to
    // complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA |= 0x10;

    return ADCW;
}

void MotorSet(unsigned int
    NomorMotor,

```

signed int
NilaiPWM)

OCR 3197 untuk

menghasilkan
sinyal 2ms >>
Servo
Mengatur
posisi untuk
sudut 90
derajat

Perhatikan bahwa timer1 dan
timer3 telah
diset
sedemikian
rupa sehingga
mempunyai
frekuensi 50Hz

1205 adalah nilai selisih
antara sudut
90 dengan 0
derajat

dengan nilai TOP pada timer =
27647 /
0x6BFF.

1123 adalah nilai selisih
antara sudut -
90 dengan 0
derajat

Dengan analogi
penskalaan,
maka
didapatkan
besar nilai

*/

OCR 862 untuk menghasilkan
sinyal 1ms >>
Servo
Mengatur
posisi untuk
sudut -90
derajat

unsigned int NilaiOCR;

if(NilaiPWM> 90){NilaiPWM=
90;} //fungsi
untuk
pembatasan
sudut
maksimal +90
derajat

OCR 1978 untuk
menghasilkan
sinyal 1.5ms
>> Servo
Mengatur
posisi untuk
sudut 0
derajat

if(NilaiPWM< 90){NilaiPWM=-
90;} //fungsi
untuk
pembatasan
sudut minimal
-90 derajat


```
if(NilaiPWM > 0)// untuk
pengkondisian
perpindahan
sudut 0 sampai
90 derajat
```

```
{
```

```
NilaiOCR= 1978 +
((float)1205*Ni
laiPWM/90);
```

```
}
```

```
else // untuk
pengkondisian
perpindahan
sudut 0 sampai
-90 derajat
```

```
{
```

```
NilaiOCR= 1978 -
((float)1123*-
NilaiPWM/90);
```

```
}
```

```
switch(NomorMotor)
```

```
{
```

```
case 0:
```

```
//Servo 1
```

```
OCR1AH=(NilaiOCR &
0b1111111100
000000)>>8;
```

```
OCR1AL=(NilaiOCR &
0b0000000011
111111);
```

```
break;
```

```
case 1:
```

```
//Servo 2
```

```
OCR1BH=(NilaiOCR&
0b1111111100
000000)>>8;
```

```
OCR1BL=(NilaiOCR&
0b0000000011
111111);
```

```
break;
```

```
default:
```

```
break;
```

```
}
```

```
void Test_Servo()
```

```
{
```

```
lcd_gotoxy(0,0);lcd_putsf("TES
Servo");
```

```
sprintf(cetak,"x=%d
y=%d",x,y);lcd_
```

```
gotoxy(0,1);lcd  
_puts(cetak);
```

```
}
```

```
if(!PB1)
```

```
{
```

```
x++; //tambahkan 1 derajat  
pada sudut  
servo vertikal
```

```
}
```

```
else if(!PB2)
```

```
{
```

```
x--; //kurangi 1 derajat pada  
sudut servo  
vertikal
```

```
}
```

```
if(!PB3)
```

```
{
```

```
y++; //tambahkan 1 derajat  
pada sudut  
servo  
horisontal
```

```
}
```

```
else if(!PB4)
```

```
{
```

```
y--; //kurangi 1 derajat pada  
sudut servo  
horisontal
```

```
if(x>=90)x=90;
```

```
else if(x<=-90)x=-90;
```

```
if(y>=90)y=90;
```

```
else if(y<=0)y=0;
```

```
MotorSet(Servo_Ver,x);
```

```
MotorSet(Servo_Hor,y);
```

```
delay_ms(100);
```

```
lcd_clear();
```

```
}
```

```
void PID_ServoX(){
```

```
float KP,KI,KD,ErrSum_Sudut;
```

```
unsigned int
```

```
DInput,LastSud  
ut_X,Max_Sud  
ut;
```

```
SP=60;
```

```
Max_Sudut=60;
```

```
KP=4;
```

```
KI=0.00;
```

```
KD=0.00;
```

```
Error= x-Nilai_X;
```

```
if(Error<=0){Error=Error*(-1);}
//agar nilai
error mutlak
positif
```

```
Konst_P = Error*KP; //ex
error=90 >>
Konst_P=90*2=
180;
```

```
if(Konst_P>Ma
x_Sudut){Konst
_P=Max_Sudut
;}
```

```
// Hitung selisih Input
(pengganti
dError pada
PID
konvensional
untuk
menghindari
bumpless
setpoint
change)
```

```
DInput = x- LastSudut_X;
```

```
LastSudut_X= x;// Hitung
integral dari
error dan
batasi nilainya.
(Sample Time
0.02 S). Batasi
juga nilai
outputnya.
```

```
ErrSum_Sudut += Error*KI*0.01;
```

```
if(ErrSum_Sudut> Max_Sudut){
ErrSum_Sudut
= Max_Sudut; }
```

```
// Hitung PID => KP*Err - KD *
dInput /
sampleTime +
ErrSum dan
batasi nilai
PIDnya
```

```
PID_Kontrol = Konst_P +
KD*DInput/0.0
2 ;
```

```
PID_Kontrol = SP-PID_Kontrol-
ErrSum_Sudut;
```

```
if(PID_Kontrol > Max_Sudut)
{ PID_Kontrol = Max_Sudut; }
```

```
if(PID_Kontrol<
=0){PID_Kontro
l=0;}
```

```
}
```

```
void PID_ServoY(){
```

```
float KP,KI,KD,ErrSum_Sudut;
```

```
unsigned int
```

```
DInput,LastSud
ut_Y,Max_Sud
ut;
```

```
SP=60;
```

```
Max_Sudut=60;
```

KP=4;

KI=0.00;

KD=0.00;

Error= y-Nilai_Y;

```
if(Error<=0){Error=Error*(-1);}
//agar nilai
error mutlak
positif
```

```
Konst_P = Error*KP; //ex
error=90 >>
Konst_P=90*2=
180;
```

```
if(Konst_P>Ma
x_Sudut){Konst
_P=Max_Sudut
;}
```

```
// Hitung selisih Input
(pengganti
dError pada
PID
konvensional
untuk
menghindari
bumpless
setpoint
change)
```

DInput = y- LastSudut_Y;

```
LastSudut_Y= y;// Hitung
integral dari
error dan
batasi nilainya.
(Sample Time
```

0.02 S). Batasi
juga nilai
outputnya.

```
ErrSum_Sudut += Error*KI*0.01;
```

```
if(ErrSum_Sudut> Max_Sudut){
ErrSum_Sudut
= Max_Sudut; }
```

```
// Hitung PID => KP*Err - KD *
dInput /
sampleTime +
ErrSum dan
batasi nilai
PIDnya
```

```
PID_Kontrol = Konst_P +
KD*DInput/0.0
2 ;
```

```
PID_Kontrol = SP-PID_Kontrol-
ErrSum_Sudut;
```

```
if(PID_Kontrol > Max_Sudut)
```

```
{ PID_Kontrol = Max_Sudut; }
```

```
if(PID_Kontrol<
=0){PID_Kontro
l=0;}
```

```
// Declare your global variables
here
```

```
//data 0 derajat = 930;
```


//data -60 derajat = 870; -15 =
892,5

//data +60 derajat = 1020; +15 =
9675;

void main(void)

{
int i;

// Declare your local variables
here

{

// Input/Output Ports initialization

// Port A initialization

// Func7=In Func6=In Func5=In

Func4=In

Func3=In

Func2=In

Func1=In

Func0=In

// State7=T State6=T State5=T

State4=T

State3=T

State2=T

State1=T

State0=T

PORTA=0x00;

DDRA=0x00;

// Port B initialization

// Func7=In Func6=In Func5=In

Func4=In

Func3=In

Func2=In

Func1=In

Func0=In

// State7=T State6=T State5=T

State4=T

State3=T

State2=T

State1=T

State0=T

PORTB=0x00;

DDRB=0x00;

// Port C initialization

// Func7=In Func6=In Func5=In

Func4=In

Func3=In

Func2=In

Func1=In

Func0=In

// State7=P State6=P State5=P

State4=P

State3=T

State2=T

State1=T

State0=T

PORTC=0xF0;

DDRC=0x00;

// Port D initialization

// Func7=In Func6=Out Func5=Out
Func4=Out
Func3=In
Func2=In
Func1=In
Func0=In

// State7=T State6=0 State5=0
State4=0
State3=T
State2=T
State1=T
State0=T

PORTD=0x00;

DDRD=0x78;

// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: Timer 0 Stopped

// Mode: Normal top=0xFF

// OC0 output: Disconnected

TCCR0=0x00;

TCNT0=0x00;

OCR0=0x00;

// Timer/Counter 1 initialization

// Clock source: System Clock

// Clock value: 1382,400 kHz

// Mode: Fast PWM top=ICR1

// OC1A output: Non-Inv.

// OC1B output: Non-Inv.

// Noise Canceler: Off

// Input Capture on Falling Edge

// Timer1 Overflow Interrupt: Off

// Input Capture Interrupt: Off

// Compare A Match Interrupt: Off

// Compare B Match Interrupt: Off

TCCR1A=0xA2;

TCCR1B=0x1A;

TCNT1H=0x00;

TCNT1L=0x00;

ICR1H=0x6B;

ICR1L=0xFF;

OCR1AH=0x00;

OCR1AL=0x00;

OCR1BH=0x00;

OCR1BL=0x00;

// Timer/Counter 2 initialization

// Clock source: System Clock

// Clock value: Timer2 Stopped

// Mode: Normal top=0xFF

// OC2 output: Disconnected

ASSR=0x00;

TCCR2=0x00;

TCNT2=0x00;

OCR2=0x00;

// External Interrupt(s)
initialization

// INT0: On

// INT0 Mode: Falling Edge

// INT1: On

// INT1 Mode: Falling Edge

// INT2: Off

GICR|=0xC0;

MCUCR=0x0A;

MCUCSR=0x00;

GIFR=0xC0;

// Timer(s)/Counter(s) Interrupt(s)
initialization

TIMSK=0x00;

// USART initialization

// Communication Parameters: 8
Data, 1 Stop,
No Parity

// USART Receiver: On

// USART Transmitter: On

// USART Mode: Asynchronous

// USART Baud Rate: 9600

UCSRA=0x00;

UCSRB=0x98;

UCSRC=0x06;

UBRRH=0x00;

UBRRL=0x47;

// Analog Comparator initialization

// Analog Comparator: Off

// Analog Comparator Input
Capture by
Timer/Counter
1: Off

ACSR=0x80;

ries|Alphanum
eric LCD menu:

SFIOR=0x00;

// RS - PORTB Bit 0

// ADC initialization

// RD - PORTB Bit 1

// ADC Clock frequency: 691,200
kHz

// EN - PORTB Bit 2

// ADC Voltage Reference: AREF
pin

// D4 - PORTB Bit 4

ADMUX=ADC_VREF_TYPE & 0xff;

// D5 - PORTB Bit 5

ADCSRA=0x84;

// D6 - PORTB Bit 6

// D7 - PORTB Bit 7

// SPI initialization

// Characters/line: 16

lcd_init(16);

// SPI disabled

// Global enable interrupts

SPCR=0x00;

#asm("sei")

// TWI initialization

}

// TWI disabled

x=0;y=0;

TWCR=0x00;

BUZZER=1;

delay_ms(400);BUZZER=0;delay_m
s(300);

// Alphanumeric LCD initialization

SHOOT=0;

// Connections are specified in the

while (1)

// Project|Configure|C

{

Compiler|Libra

if(!PB1){


```
lcd_clear();
```

```
while(1){
```

```
    //proses memasukkan input  
    nilai x dan y
```

```
    Proses_Parsing=1;
```

```
    //NilaiX adalah nilai untuk  
    sudut servo  
    vertikal
```

```
    //NilaiY adalah nilai untuk  
    sudut servo  
    Horizontal
```

```
    if(x==Nilai_X &&  
        y==Nilai_Y){
```

```
        MotorSet(Servo_Ver,x);
```

```
        MotorSet(Servo_Hor,y);
```

```
    }
```

```
    else if (x!=Nilai_X ||  
        y!=Nilai_Y){del  
        ay_ms(100);
```

```
    while(x!=Nilai_X ||  
        y!=Nilai_Y)
```

```
    {
```

```
        if(Nilai_X  
            ==x){MotorSet(  
                Servo_Ver,x);}
```

```
        else if(Nilai_X <0){
```

```
            if(x==Nilai_X){  
                MotorSet(Servo_Ver,x);}
```

```
            else if(Nilai_X <x){
```

```
                Proses_Parsing=0;
```

```
                pot_x=read_ad  
                c(0);pot_y=rea  
                d_adc(1);
```

```
                sprintf(cetak,"F  
                x=%d  
                Fy=%d",pot_x,  
                pot_y);lcd_got  
                oxy(0,1);lcd_pu  
                ts(cetak);
```

```
                x--  
                ;MotorSet(Servo_Ver,x);PID_S  
                ervoX();delay_  
                ms(PID_Kontro  
                l);
```

```
                delay_ms(0);
```

```
            }
```

```
            else if(Nilai_X >x){
```

```
                Proses_Parsing=0;
```

```
                pot_x=read_ad  
                c(0);pot_y=rea  
                d_adc(1);
```

```
sprintf(cetak,"F
x=%d
Fy=%d",pot_x,
pot_y);lcd_got
oxy(0,1);lcd_pu
ts(cetak);
```

```
x++;MotorSet(
Servo_Ver,x);PI
D_ServoX();del
ay_ms(PID_Ko
ntrol);
```

```
delay_ms(0);
```

```
}
```

```
Proses_Parsing=1;
```

```
}
```

```
else if (Nilai_X>=0){
```

```
if(x==Nilai_X){
MotorSet(Serv
o_Ver,x);}
```

```
else if(Nilai_X >x){
```

```
Proses_Parsing=0;
```

```
pot_x=read_ad
c(0);pot_y=rea
d_adc(1);
```

```
sprintf(cetak,"F
x=%d
```

```
Fy=%d",pot_x,
pot_y);lcd_got
oxy(0,1);lcd_pu
ts(cetak);
```

```
x++;MotorSet(
Servo_Ver,x);PI
D_ServoX();del
ay_ms(PID_Ko
ntrol);
```

```
delay_ms(0);
```

```
}
```

```
else if(Nilai_X < x){
```

```
Proses_Parsing=0;
```

```
pot_x=read_ad
c(0);pot_y=rea
d_adc(1);
```

```
sprintf(cetak,"F
x=%d
Fy=%d",pot_x,
pot_y);lcd_got
oxy(0,1);lcd_pu
ts(cetak);
```

```
x--
```

```
;MotorSet(Serv
o_Ver,x);PID_S
ervoX();delay_
ms(PID_Kontro
l);
```

```
delay_ms(0);
```

}

Proses_Parsing=1;

}

if(Nilai_Y
==y){MotorSet(
Servo_Hor,y);}

else if(Nilai_Y<0){y=0;}

else if (Nilai_Y>=0){

if(y==Nilai_Y){
MotorSet(Serv
o_Hor,y);}

else if(Nilai_Y > y){

Proses_Parsing=0;

pot_x=read_ad
c(0);pot_y=rea
d_adc(1);

sprintf(cetak,"F
x=%d
Fy=%d",pot_x,
pot_y);lcd_got
oxy(0,1);lcd_pu
ts(cetak);

y++;MotorSet(
Servo_Hor,y);d
elay_ms(0);PID

_ServoY());dela
y_ms(PID_Kont
rol);}

else if(Nilai_Y<y){

Proses_Parsing=0;

pot_x=read_ad
c(0);pot_y=rea
d_adc(1);

sprintf(cetak,"F
x=%d
Fy=%d",pot_x,
pot_y);lcd_got
oxy(0,1);lcd_pu
ts(cetak);

y--

;MotorSet(Serv
o_Hor,y);delay
_ms(0);PID_Ser
voY());delay_ms
(PID_Kontrol);}

Proses_Parsing=1;

}

}

if(x==Nilai_X &&
y==Nilai_Y){

pot_x=1.5*x+8
80;pot_y=1*y+
900;

```
//pot_x=read_
adc(0);pot_y=r
ead_adc(1);

sprintf(cetak,"Fx=%3d
Fy=%3d
",pot_x,pot_y);
lcd_gotoxy(0,0)
;lcd_puts(cetak
);
```

```
MotorSet(Servo
o_Ver,x);Motor
Set(Servo_Hor,
y);lcd_gotoxy(0
,1);lcd_putsf("P
repare
Shooting");dela
y_ms(2000);
```

```
SHOOT=1;
```

```
lcd_gotoxy(0,1)
;lcd_putsf("Sho
oting Target ");
```

```
delay_ms(1000);
```

```
//lcd_gotoxy(0,
1);lcd_putsf("S
hooting
Target");
```

```
//SHOOT=0;
```

```
//delay_ms(10000);
```

```
//SHOOT=1;
```

```
//lcd_gotoxy(0,
1);lcd_putsf("S
hooting
Target");
```

```
//delay_ms(1000);
```

```
//lcd_gotoxy(0,
1);lcd_putsf("S
hooting
Target");
```

```
//SHOOT=0;
```

```
//delay_ms(10000);
```

```
//SHOOT=1;
```

```
//lcd_gotoxy(0,
1);lcd_putsf("S
hooting
Target");
```

```
//delay_ms(1000);
```

```
//lcd_gotoxy(0,
1);lcd_putsf("S
hooting
Target");
```

```
//SHOOT=0;
```

```
//delay_ms(3000);
```

```
//SHOOT=1;
```

```
//lcd_gotoxy(0,
```



```
1);lcd_putsf("S  
hooting  
Target");
```

```
delay_ms(200);
```

```
//delay_ms(1000);
```

```
//lcd_gotoxy(0,  
1);lcd_putsf("S  
hooting  
Target");
```

```
//SHOOT=0;
```

```
//delay_ms(3000);
```

```
//SHOOT=1;
```

```
//lcd_gotoxy(0,  
1);lcd_putsf("S  
hooting  
Target");
```

```
//delay_ms(1000);
```

```
//lcd_gotoxy(0,  
1);lcd_putsf("S  
hooting  
Target");
```

```
//SHOOT=0;
```

```
//delay_ms(5000);
```

```
lcd_gotoxy(0,1)  
;lcd_putsf("Dat  
a Input  
Ready");
```

```
}
```

```
}
```

```
}
```

```
}
```

```
else if (!PB2){
```

```
x=0;y=0;
```

```
lcd_clear();
```

```
lcd_gotoxy(0,0)  
;lcd_putsf("Pre  
paring Test");
```

```
for(i=0;i<=10;i++)
```

```
{lcd_gotoxy(i,1  
);lcd_putsf(">")  
;delay_ms(100  
);}
```

```
lcd_clear();
```

```
while(1){
```

```
//proses test sudut servo  
dengan input  
tombol
```

Test_Servo();//merupakan
fungsi test
servo yang
sudah
dideklarasikan
di fungsi di atas

}

}

else

{

MotorSet(Servo_Ver,x);

MotorSet(Servo_Hor,y);

lcd_gotoxy(0,0)

;lcd_putsf("S2=
Auto System");

lcd_gotoxy(0,1)

;lcd_putsf("S3=
Tes System");

}

}

}

BIODATA PENULIS



Irvan Dwi Putra, dilahirkan di Tasikmalaya pada tanggal 29 Oktober 1990. Penulis adalah putra kedua dari tiga bersaudara dari pasangan Bapak Muhamad Kusna dan Ibu Rohayati. Pendidikan dasar ditempuh di SD Cibuluh 1 Bogor, lulus pada tahun 2003. Pendidikan menengah pertama ditempuh di SMP Negeri 1 Bogor, lulus pada tahun 2006. Pendidikan menengah atas ditempuh di SMA Negeri 1 Bogor, lulus pada tahun 2009. Pada tahun yang sama, penulis melanjutkan pendidikan di Jurusan Teknik Mesin Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Di Jurusan Teknik Mesin ini, penulis mengambil Bidang Studi Teknik Manufaktur.

RANCANG BANGUN SISTEM KONTROL PELONTAR PELURU DENGAN DUA SUMBU PUTAR

Irvan Dwi Putra dan Arif Wahjudi

Jurusan Teknik Mesin, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

wahjudi.its@gmail.com

Abstrak-- Tuntutan terhadap minimnya korban dalam penggunaan teknologi persenjataan menjadi hal utama yang ingin dicapai dalam bidang pertahanan. Hal ini menuntut badan pertahanan negara Republik Indonesia untuk selalu berinovasi dalam menciptakan suatu teknologi yang aman dan memiliki tingkat akurasi yang tinggi. Untuk itu telah dilengkapi oleh Pemerintah Indonesia sebuah alat utama sistem pertahanan dengan persenjataan otomatis dan mampu dikendalikan dari jarak jauh, salah satu alat persenjataan otomatis yang digunakan oleh badan pertahanan negara kita adalah senjata semi otomatis. Pada tugas akhir ini dirancang dan dibuat sebuah prototype pelontar peluru yang dilengkapi dengan dua sumbu putar. Prototype sistem kontrol diuji dengan bantuan software CodeVisionAVR dan software EAGLE. Hasil program yang telah dirancang didownload ke dalam mikrokontroler menggunakan downloader. Mikrokontroler tersebut yang akan menggerakkan motor servo. Pada tugas akhir ini akan dibuat suatu sistem kontrol yang dapat menggerakkan motor servo bergerak inklinasi $+60^{\circ}$ dan bergerak rotasi $\pm 60^{\circ}$ dengan presisi. Sistem kontrol diuji untuk menggerakkan pelontar peluru sesuai dengan kordinat yang di inputkan. Hasil akhir ini adalah rancang bangun sistem pelontar peluru yang terdiri dari mikrokontroler hasil perakitan, dua motor servo dengan nilai torsi maksimal sebesar 3.7 kg.cm, dua potensiometer, pengatur relay, LCD pembaca nilai potensiometer. Selanjutnya dari percobaan yang dilakukan, sistem kontrol pelontar peluru dapat dikendalikan, tetapi masih terjadi noise yang menyebabkan nilai toleransi akurasi yang berbeda. Nilainya berada diantara $\pm 0,30^{\circ}$ dan berada diluar $\pm 0,8^{\circ}$.

Kata kunci-- pelontar peluru plastik, kestabilan sistem, unit kontrol, mikrokontroler.

I. PENDAHULUAN

Perkembangan teknologi selaras dengan perkembangan jaman, salah satunya dibidang pertahanan. Tuntutan terhadap minimnya korban dalam penggunaan teknologi persenjataan menjadi hal utama yang ingin dicapai. Hal ini menuntut badan pertahanan suatu negara untuk selalu berinovasi dalam menciptakan suatu teknologi yang aman dan memiliki tingkat akurasi yang tinggi. Oleh karena itu pemerintah telah melengkapi alat utama sistem pertahanan dengan persenjataan yang bersifat otomatis dan mampu dikendalikan dari jarak jauh. Senapan semi otomatis adalah salah satu alat persenjataan otomatis yang digunakan oleh badan pertahanan negara kita.

Saat ini telah banyak pengembangan senjata-senjata pengenalan target secara otomatis. Salah satu penelitian yang berhubungan dengan senjata pengenalan target secara otomatis[1]. Mereka meneliti berbagai macam aplikasi metode

kontrol konvensional pada *automatic turret gun*. Selain itu terdapat penelitian tentang pengenalan target [2]. Mereka meneliti pemanfaatan sinyal *monogenic* untuk mendapatkan karakteristik pencitraan SAR (*synthetic aperture radar*) dengan teori DS (*Dempster-Shafer*). Dalam mengenali target melalui agregat informasi, memberikan hasil metode dalam metode ini yang jauh lebih efektif untuk menangani kondisi operasi yang berlebihan, seperti konfigurasi yang berbeda dan depresi. Selain itu paper tentang sistem kontrol otomatis pada robot telah dikembangkan [3]. Sistem *ballast* yang dikembangkan dikontrol dengan penghantar kabel sehingga masukan yang diterima oleh mikrokontroler. Robot bawah air dapat menyelam di dalam air sesuai kedalaman yang diinginkan dan mengapung ke kondisi awal dengan menggunakan sistem *ballast*.

Dari beberapa penelitian yang telah ada, kita sebagai warga Negara Indonesia harus berperan aktif dalam membantu setiap pembangunan. Salah satunya adalah pertahanan Negara. Dengan membantu menciptakan alat senjata pengenalan otomatis yang sudah berkembang. Setiap persenjataan harus memiliki tingkat akurasi dan kepresisian. Agar rancangan ini sesuai dengan kebutuhan bidang pertahanan, maka dibuat suatu pemodelan *prototype* dengan menggunakan pelontar peluru plastik. Dari *prototype* ini dilakukan pengujian perancangan sistem kontrol dengan bantuan *software* CodeVisionAVR pada laptop untuk menggerakkan pelontar peluru plastik sesuai dengan kordinat yang kita input. Hasil program yang telah dirancang didownload ke dalam mikrokontroler menggunakan downloader. Mikrokontroler tersebut yang akan menggerakkan motor servo. Pada tugas akhir ini akan dibuat suatu sistem kontrol yang dapat menggerakkan motor servo bergerak naik, bergerak ke kiri dan ke kanan dengan cepat sehingga dapat mempercepat proses kerja motor servo tersebut.

II. METODOLOGI PENELITIAN

1. Studi literatur dan Lapangan

Studi literatur dilakukan untuk memperoleh materi yang berkaitan dengan sistem kontrol pelontar peluru dengan dua sumbu putar. Literatur yang dikaji berasal dari jurnal dan buku.

2. Perumusan masalah

Langkah ini dilakukan untuk menentukan permasalahan yang akan dikaji dan dicari solusi terbaiknya. Dalam Tugas

Akhir ini rumusan masalahnya adalah bagaimana merancang perangkat lunak sistem kontrol yang akan diaplikasikan pada pelontar peluru dengan dua sumbu putar.

3. Perancangan dan pemilihan komponen

Pada tahap ini dilakukan perancangan sistem kontrol. Perancangan tersebut pemilihan komponen yang akan digunakan pada sistem. Pemilihan ini disesuaikan dengan kebutuhan sistem yang akan dibuat. Contohnya sensor memiliki spesifikasi sesuai target pengukuran, jenis motor memiliki torsi yang cukup untuk menggerakkan beban, mikrokontroler memiliki spesifikasi sesuai kebutuhan dan sebagainya.

4. Simulasi rancangan menggunakan *software*

Pada tahap ini dilakukan perakitan komponen menjadi perangkat keras atau *hardware* sesuai rancangan pada *software* EAGLE. Selanjutnya dibuat program menggunakan *software* CodeVisionAVR yang akan digunakan pada mikrokontroler agar dapat menjalankan fungsinya. Perangkat keras ini kemudian diuji agar dapat berfungsi dengan baik.

5. Resolusi $< \pm 0,38^\circ$.

Setelah perancangan berupa desain perangkat keras dengan komponen yang telah dipilih selesai, perangkat keras diatur agar resolusi kecermatan sistem tidak melebihi $\pm 0,38^\circ$.

6. Pengujian

Langkah ini dilakukan untuk mengetahui bahwa pembuatan sistem kontrol telah memenuhi target.

7. Kepresisian = resolusi

Sistem kontrol memenuhi target jika hasil kecermatan pembacaan alat ukur sama dengan $\pm 0,38^\circ$.

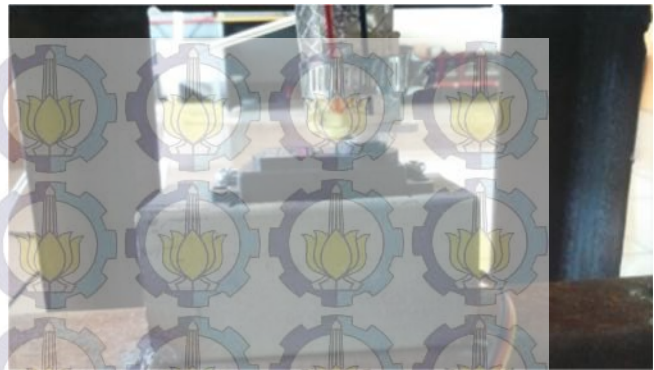
8. Kesimpulan dan saran

Kesimpulan dari penelitian Tugas Akhir dapat diambil jika sistem kontrol telah bekerja dengan baik. Kesimpulan berupa nilai error yang didapat dari pengujian sistem kontrol pelontar peluru dengan dua sumbu putar. Saran berupa masukan untuk pengembangan sistem kontrol pelontar peluru dengan dua sumbu putar pada khususnya dan perangkat keras unit kontrol lain pada umumnya.

III. HASIL DAN PEMBAHASAN

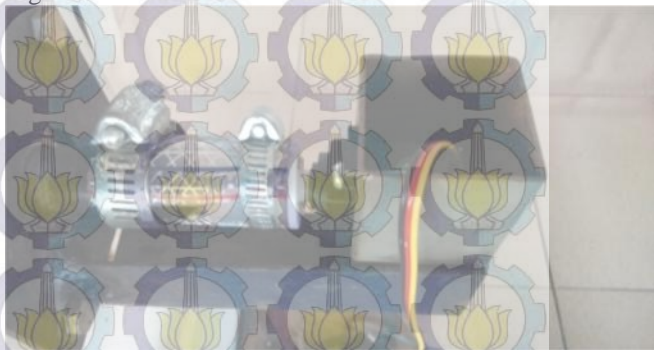
A. Perakitan Sistem

Motor servo diletakkan pada dudukan yang telah dibuat. Dudukan ini dihubungkan dengan kaki meja alat pelontar peluru menggunakan rumah motor yang di las. Pada Gambar 1 dan Gambar 2 terlihat poros motor telah terhubung dengan poros pelontar peluru menggunakan *fleksibel joint*. Jika diberi tegangan dan sinyal dari driver maka motor servo berputar dan menggerakkan poros.



Gambar 1. Susunan Motor Servo arah Rotasi, Rumah Dudukan dan Fleksibel Joint

Motor servo ini memiliki spesifikasi 0.19 detik per 60° pada saat tidak ada beban. Torsi maksimumnya sebesar 3kg.cm pada saat diberikan daya 4.8V. Sudut maksimum yang bisa ditempuh sebesar 180° . Motor servo ini digunakan untuk bergerak rotasi $\pm 60^\circ$.



Gambar 2. Susunan Motor Servo arah Inklinasi, Rumah Dudukan dan Fleksibel Joint

Motor servo ini memiliki spesifikasi 0.19 detik per 60° pada saat tidak ada beban. Torsi maksimumnya sebesar 3kg.cm pada saat diberikan daya 4.8V. Sudut maksimum yang bisa ditempuh sebesar 180° . Motor servo ini digunakan untuk bergerak inklinasi $+60^\circ$. Potensiometer terletak pada bagian dalam motor servo. Fungsi potensiometer pada alat pelontar peluru adalah sebagai hambatan dengan nilai ADC maksimal 1024. pada sistem ini nilai potensiometer ditentukan dengan melakukan kalibrasi antara sudut dan nilai pada potensiometer. Pada potensiometer motor servo arah rotasi, nilai potensiometer didapat sebesar 790 pada sudut -60° , 880 pada sudut 0° dan 970 pada sudut $+60^\circ$. Sedangkan pada potensiometer motor servo arah inklinasi nilai potensiometer didapat 900 pada sudut 0° dan 960 pada sudut 60° . Dengan menggunakan persamaan :

$$x = m.z + c$$

(1) Untuk potensiometer pada motor servo arah rotasi.

$$y = m.z + c$$

Untuk potensiometer pada motor servo arah inklinasi.

Dimana :

x = nilai potensiometer pada arah rotasi

y = nilai potensiometer pada arah inklinasi

m = nilai gradient

c = nilai konstanta

z = sudut

(2)

Contoh perhitungan menentukan nilai potensiometer pada motor servo arah rotasi sudut -60° , sudut 0° dan sudut $+60^\circ$ dengan menggunakan persamaan 1.

- Menentukan nilai c pada sudut 0°

$$x = m \cdot x + c$$

$$880 = m(0^\circ) + c$$

$$880 = c, \text{ nilai } c \text{ adalah } 880$$

- Menentukan nilai m pada sudut $+60^\circ$

$$x = m \cdot z + c$$

$$970 = m(+60^\circ) + 880$$

$$90 = m(+60^\circ)$$

$$m = 1 \frac{1}{2}$$

sehingga didapatkan :

$$x = 1 \frac{1}{2} \cdot z + 880 \quad (3)$$

Contoh perhitungan menentukan nilai potensiometer pada motor servo arah inklinasi sudut 60° dengan menggunakan persamaan 2.

- Menentukan nilai c pada sudut 0°

$$y = m \cdot z + c$$

$$900 = m(0^\circ) + c$$

$$900 = c, \text{ nilai } c \text{ adalah } 900$$

- Menentukan nilai m pada sudut 60°

$$y = m \cdot z + c$$

$$960 = m(60^\circ) + 900$$

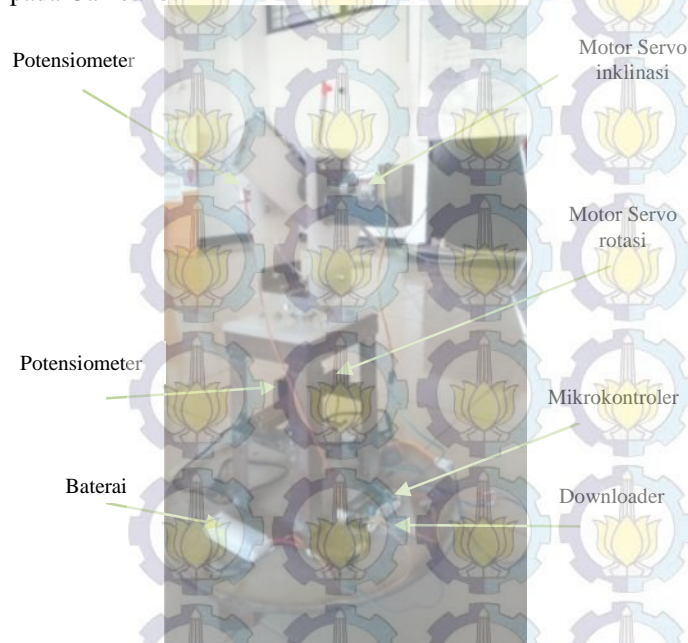
$$60 = m(60^\circ)$$

$$m = 1$$

sehingga didapatkan :

$$y = 1 \cdot z + 900 \quad (4)$$

Unit kontrol yang telah tersusun sesuai skema dapat dilihat pada Gambar 3.



Gambar 3. Susunan Hardware Pelontar Peluru

Dari gambar tersebut dapat dilihat bahwa letak dan sambungan tiap komponen sudah sesuai ketentuan fungsi masing-masing. Penyusunan kabel penghubung juga diatur agar tidak menghambat gerak pelontar peluru. Dengan demikian perangkat keras (*hardware*) pelontar peluru dapat berfungsi dengan baik.

B. Algoritma Sistem

Pada proses pembuatan perangkat lunak diperlukan suatu perencanaan yang berupa algoritma-algoritma pengolahan citra supaya tujuan akhir dari perangkat lunak ini tercapai. Algoritma umum dari perangkat lunak ini yaitu:

1. Sistem pada kondisi diam dan semua perangkat keras telah tersambung dengan mikrokontroler.
2. Tekan tombol *switch on/off* untuk menyalakan mikrokontroler. Pada saat ini kompresor dalam keadaan menyala, dan solenoid valve dalam keadaan mati.
3. Untuk menginputkan data dari komputer terlebih dahulu dilakukan penekanan pada push button S2. Pada penekanan tersebut *push button* mengirimkan kode yang memungkinkan menginput data via komputer.
4. Selanjutnya adalah menginputkan data berupa sudut. Data tersebut masuk ke mikrokontroler kemudian diteruskan ke motor servo. Motor servo bergerak ke arah koordinat yang diinputkan, jika sudut sesuai dengan pembacaan busur drajat maka input sudut berhasil.
5. Setelah motor servo bergerak menuju target, perintah selanjutnya dari program adalah menyalakan *solenoid valve* yang akan meneruskan tekanan dari kompresor ke laras pelontar peluru untuk mendorong peluru plastik keluar dari laras.
6. Solenoid valve hanya menyala untuk beberapa saat, setelah itu mati kembali. Setelah proses pelontar pertama selesai, kompresor menyala kembali untuk mengisi tekanan yang hilang.

Hardware pada system ini terdiri dari *Power Supply*, *Minimum Sistem*, *Pushbutton*, *Relay*, *LCD*, dan juga motor servo. Akan dilakukan pengujian sebanyak empat kali pada tiga koordinat sasaran tembak yang ditentukan.

C. Pengujian dan Analisa Hasil Pengujian

Untuk pengujian sudut yang ditentukan sebesar $(-60^\circ, 30^\circ)$, pengujian selanjutnya sebesar $(0^\circ, 30^\circ)$ dan setelah itu pengujian terakhir sebesar $(+60^\circ, 30^\circ)$. Pengujian tersebut dilakukan sebanyak empat kali. Mengulang proses 5 dan 6 untuk pelontaran peluru hingga 5 kali pelontaran. Selanjutnya kembali mengulang proses 4-7 dengan input sudut yang berbeda. Percobaan dilakukan sebanyak empat kali pada tiga titik yang ditentukan.

- Analisa Hasil Pengujian Percobaan Pada Titik (-60°,30°)

Dari Pengujian Percobaan Pada Titik (-60°,30°) sebanyak empat kali didapat nilai ADC hasil pembacaan potensiometer pada tabel 1.

Tabel 1 Tabel perhitungan nilai rata-rata

Percobaan Ke-	X(-60°)	Y(30°)
1	790	930
2	789	931
3	790	932
4	792	930

Perhitungan rata-rata (x)

$$\bar{x} = \sum \frac{x_i}{n} = \frac{x_1 + x_2 + x_3 + x_4}{n} = \frac{790 + 789 + 790 + 792}{4} \approx 790.25$$

Perhitungan rata-rata (y)

$$\bar{y} = \sum \frac{y_i}{n} = \frac{y_1 + y_2 + y_3 + y_4}{n} = \frac{930 + 931 + 932 + 930}{4} = 930.85$$

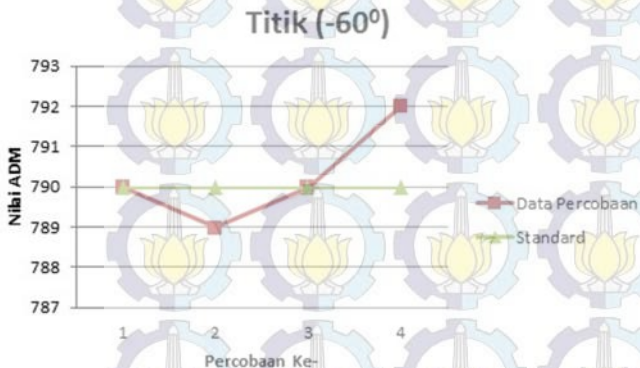
Dari perhitungan nilai rata-rata, didapatkan nilai x sebesar 790.25 dan nilai \bar{y} sebesar 930.85. Dari nilai tersebut dikonversikan menjadi sudut dengan persamaan rumus 3 untuk nilai x dan persamaan rumus 4 nilai \bar{y} . Untuk nilai x,

$$z = \frac{790.25 - 880}{1 \frac{1}{2}} \\ z = -59.833^\circ$$

Untuk nilai y,

$$z = \frac{930.58 - 900}{1} \\ z = 30.58^\circ$$

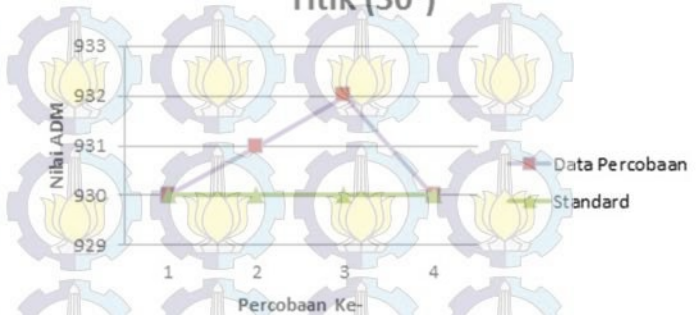
Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik (-60°,30°).



Gambar 4. Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik (-60°)

Dari grafik perbandingan data percobaan dengan data acuan dan hasil perhitungan nilai rata-rata, didapatkan data hasil percobaan sesuai dengan data acuan.

Titik (30°)



Gambar 5. Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik (30°)

Dari grafik perbandingan data percobaan dengan data acuan dan hasil perhitungan nilai rata-rata, didapatkan data hasil percobaan tidak sesuai dengan data acuan.

- Analisa Hasil Pengujian Percobaan Pada Titik (0°,30°)

Dari Pengujian Percobaan Pada Titik (0°,30°) sebanyak empat kali didapat nilai ADC hasil pembacaan potensiometer pada tabel 2.

Tabel 2 Tabel perhitungan nilai rata-rata

Percobaan Ke-	X(0°)	Y(30°)
1	881	930
2	880	931
3	880	930
4	882	930

Perhitungan rata-rata (x)

$$\bar{x} = \sum \frac{x_i}{n} = \frac{x_1 + x_2 + x_3 + x_4}{n} = \frac{881 + 880 + 880 + 882}{4} \approx 880.75$$

Perhitungan rata-rata (y)

$$\bar{y} = \sum \frac{y_i}{n} = \frac{y_1 + y_2 + y_3 + y_4}{n} = \frac{930 + 931 + 930 + 930}{4} = 930.25$$

Dari perhitungan nilai rata-rata, didapatkan nilai x sebesar 880.75 dan nilai \bar{y} sebesar 930.25. Dari nilai tersebut dikonversikan menjadi sudut dengan persamaan rumus 4.3 untuk nilai x dan persamaan rumus 4.4 nilai \bar{y} .

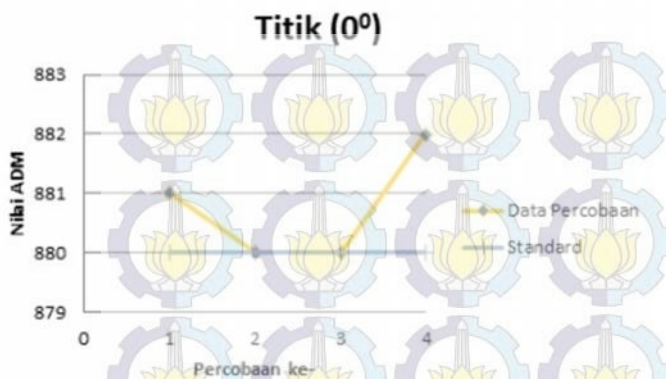
Untuk nilai x,

$$z = \frac{880.75 - 880}{1 \frac{1}{2}} \\ z = 0.5^\circ$$

Untuk nilai y,

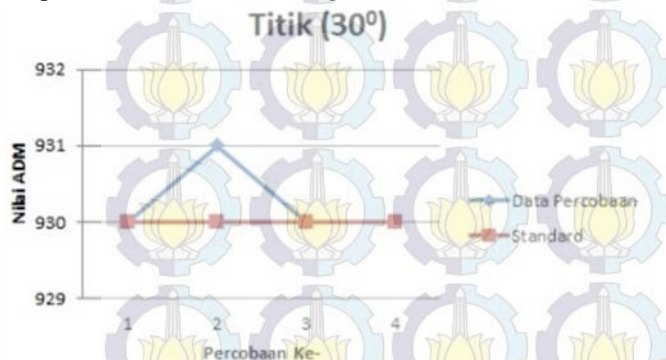
$$z = \frac{930.25 - 900}{1} \\ z = 30.25^\circ$$

Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik (0°,30°).



Gambar 6. Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik (0°)

Dari grafik perbandingan data percobaan dengan data acuan dan hasil perhitungan nilai rata-rata, didapatkan data hasil percobaan tidak sesuai dengan data acuan.



Gambar 7. Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik (30°)

Dari grafik perbandingan data percobaan dengan data acuan dan hasil perhitungan nilai rata-rata, didapatkan data hasil percobaan sesuai dengan data acuan.

- **Analisa Hasil Pengujian Percobaan Pada Titik (+60°,30°)**

Dari Pengujian Percobaan Pada Titik (+60°,30°) sebanyak empat kali didapat nilai ADC hasil pembacaan potensiometer pada tabel 3.

Tabel 3 Tabel perhitungan nilai rata-rata

Percobaan Ke-	X(+60°)	Y(30°)
1	971	930
2	972	929
3	970	931
4	970	930

Perhitungan rata-rata (\bar{x})

$$\bar{x} = \sum \frac{x_i}{n} = \frac{x_1 + x_2 + x_3 + x_4}{n} = \frac{971 + 972 + 970 + 970}{4} = 970,75$$

Perhitungan rata-rata (\bar{y})

$$\bar{y} = \sum \frac{y_i}{n} = \frac{y_1 + y_2 + y_3 + y_4}{n} = \frac{930 + 929 + 931 + 930}{4} = 930$$

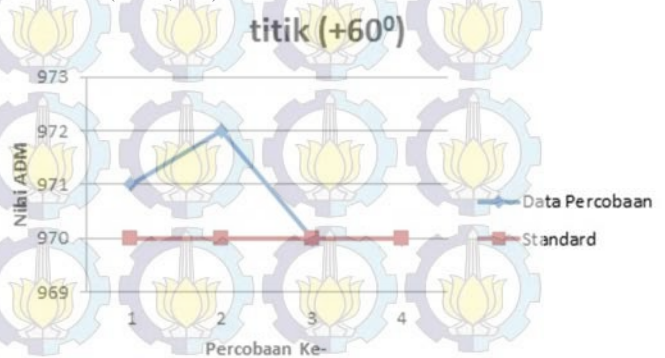
Dari perhitungan nilai rata-rata, didapatkan nilai \bar{x} sebesar 970.75 dan nilai \bar{y} sebesar 930. Dari nilai tersebut dikonversikan menjadi sudut dengan persamaan rumus 3 untuk nilai x dan persamaan rumus 4 nilai \bar{y} . Untuk nilai x ,

$$z = \frac{970,75 - 880}{1\frac{1}{2}} = 60,5^\circ$$

Untuk nilai y ,

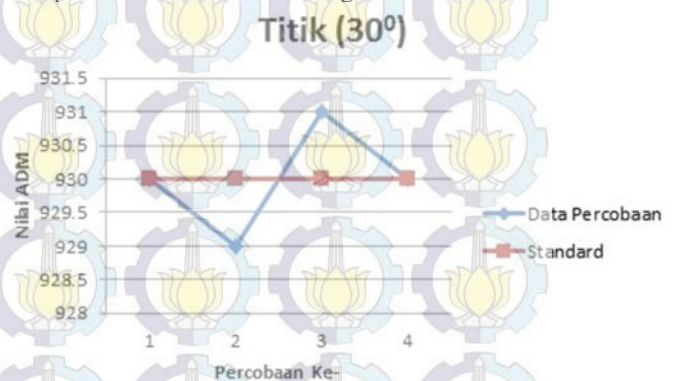
$$z = \frac{930 - 900}{1} = 30^\circ$$

Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik (+60°,30°).



Gambar 8. Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik (+60°)

Dari grafik perbandingan data percobaan dengan data acuan dan hasil perhitungan nilai rata-rata, didapatkan data hasil percobaan tidak sesuai dengan data acuan.



Gambar 9. Grafik Perbandingan Data Percobaan Dengan Data Acuan Pada Titik (30°)

Dari grafik perbandingan data percobaan dengan data acuan dan hasil perhitungan nilai rata-rata, didapatkan data hasil percobaan sesuai dengan data acuan.

IV. KESIMPULAN

Perancangan sistem kontrol pelontar peluru plastik sesuai dengan rancangan. Dari ketiga pengujian tersebut dapat dilihat pembacaan sudut dengan bantuan potensiometer terdapat beberapa variasi nilai. Nilai-nilai tersebut ada yang berada di

dalam *range* $\pm 0.38^\circ$ (nilai z -59.833° pada sudut -60° , nilai z 30.25° pada sudut 30° titik kedua dan nilai z 30° pada sudut 30° titik ketiga) dan ada yang berada diluar *range* $\pm 0.38^\circ$ (nilai z 30.58° pada sudut 30° titik pertama, nilai z 0.5° pada sudut 0° dan nilai z 60.5° pada sudut 60°). Nilai yang berada diluar *range* tersebut dikarenakan terdapat beberapa *noise*. *Noise* yang terjadi berasal dari proses pemasangan mekanik pelontar peluru yang kurang teliti sehingga menyebabkan sensor potensiometer tidak berkerja maksimal dalam membaca data.

DAFTAR PUSTAKA

- [1] Naszir, T., Pramujati, B., Nurhadi, H., dan Pitowarno, E., 2014. "Control Simulation of An Automatic Turret Gun Based on Force Control Method", *IEEE Xplore Digital Library* 1,1:13-18.
- [2] Dong, G., dan Kuang, G. 2014. "Target Recognition via Information Aggregation Through Dempster-Shafer's Evidence Theory", *IEEE Geosci, Remote Sens. Lett.*, 12,1:1247-1251.
- [3] Chandra, Y., Purnomo, Didik S., dan Suryawati N., Endah. 2011. "Rancang Bangun Sistem Ballast Pada ROV (Remotely Underwater Vehicle)", Jurusan Teknik Mekatronika, Kampus PENS-ITS.
- [4] Nasution, F., 2011. "Perancangan Telemetri Suhu Ruang berbasis Mikrokontroler ATMega", Universitas Sumatera Utara.
- [6] Andri, M. November. 2013. *Pulse Width Modulation (PWM)*, http://andri_mz.staff.ipb.ac.id/pulse-width-modulation-pwm/.
- [7] Nise, N., 2011. "Control Systems Engineering Sixth Edition", California State Polytechnic University, Pomona.



SIDANG TUGAS AKHIR



**LAB PERANCANGAN
ENGENGBANGAN
RODUK**
TEKNIK MESIN - INSTITUT TEKNOLOGI SEPULUH NOPEMBER

RANCANG BANGUN SISTEM KONTROL PELONTAR PELURU DENGAN DUA SUMBU PUTAR

**Oleh:
Irvan Dwi Putra
2109 100 117**

**Dosen Pembimbing:
Arif Wahyudi ST., MT., Ph.D.**



**JURUSAN TEKNIK MESIN
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2015**

LATAR BELAKANG



**PERANCANGAN
ENGEMBANGAN
PRODUK**
TEKNIK MESIN - INSTITUT TEKNOLOGI SEPULUH NOPEMBER



TM



Sejarah
Persenjataan



Penelitian Terdahulu



Nasyir dkk. (2014)

Judul

: *Control Simulation of An Automatic Turret Gun Based on Force Control Method*

Metode

: kontrol PID, RAC dan RACAFC

Variabel proses

: variasi untuk berbagai macam sudut baik di *azimuth* dan elevasi sumbu *turret gun*

Hasil

**: ~RACAFC menunjukkan performa yang lebih baik dari kedua metode kontrol ketika diberi beban
~Akan tetapi jika percobaan tanpa pemberian beban, hasil yang memuaskan didapat dari RAC dan RACAFC**





Penelitian Terdahulu



Chandra dkk. (2011)

Judul

: Rancang Bangun Sistem *Ballast* Pada ROV (*Remotely Underwater Vehicle*)

Metode

: sistem *Ballast*

Variabel proses

: sistem *ballast* dirancang dengan sistem pompa air

Hasil

: ~ Robot bawah air dapat menyelam di dalam air sesuai kedalaman yang diinginkan dan mengapung ke kondisi awal dengan menggunakan sistem *ballast*.





Rumusan Masalah

Bagaimanakah perancangan sistem kontrol pelontar peluru yang mempunyai dua sumbu putar?

Bagaimana membangun sistem kontrol pelontar peluru dengan kecermatan $\pm 0.38^\circ$?

Batasan Masalah

Jarak inklinasi $+60^\circ$ pada motor servo 1.

Jarak rotasi $\pm 60^\circ$ pada motor servo 2.

Menggunakan bantuan software CodeVisionAVR, EAGLE

Menggunakan sistem kontrol IC ATmega 32.



Tujuan Penelitian

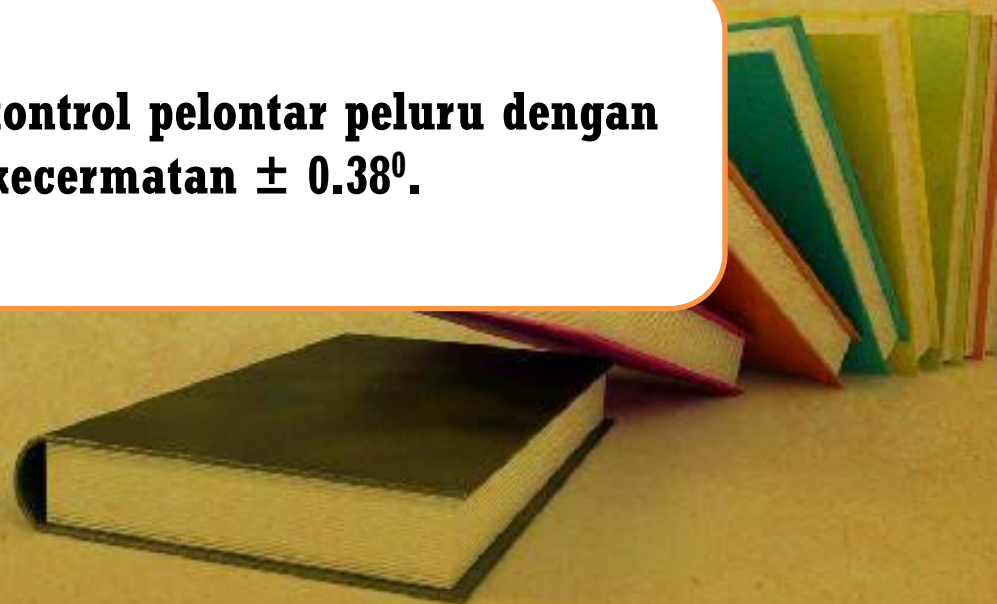


1

Perancangan sistem kontrol pelontar peluru dengan dua sumbu putar.

2

Membangun sistem kontrol pelontar peluru dengan toleransi kecermatan $\pm 0.38^\circ$.





Manfaat Penelitian



Memperkaya dan memperdalam wacana dalam bidang ilmu sistem kontrol, terutama pada perancangan sistem kontrol pelontar peluru.

Mesin pelontar peluru yang telah berfungsi dapat digunakan untuk penelitian lebih lanjut.

Memberikan kontribusi nyata pada pengembangan teknologi perancangan pelontar peluru, demi terciptanya kemajuan teknologi nasional khususnya di bidang pertahanan.

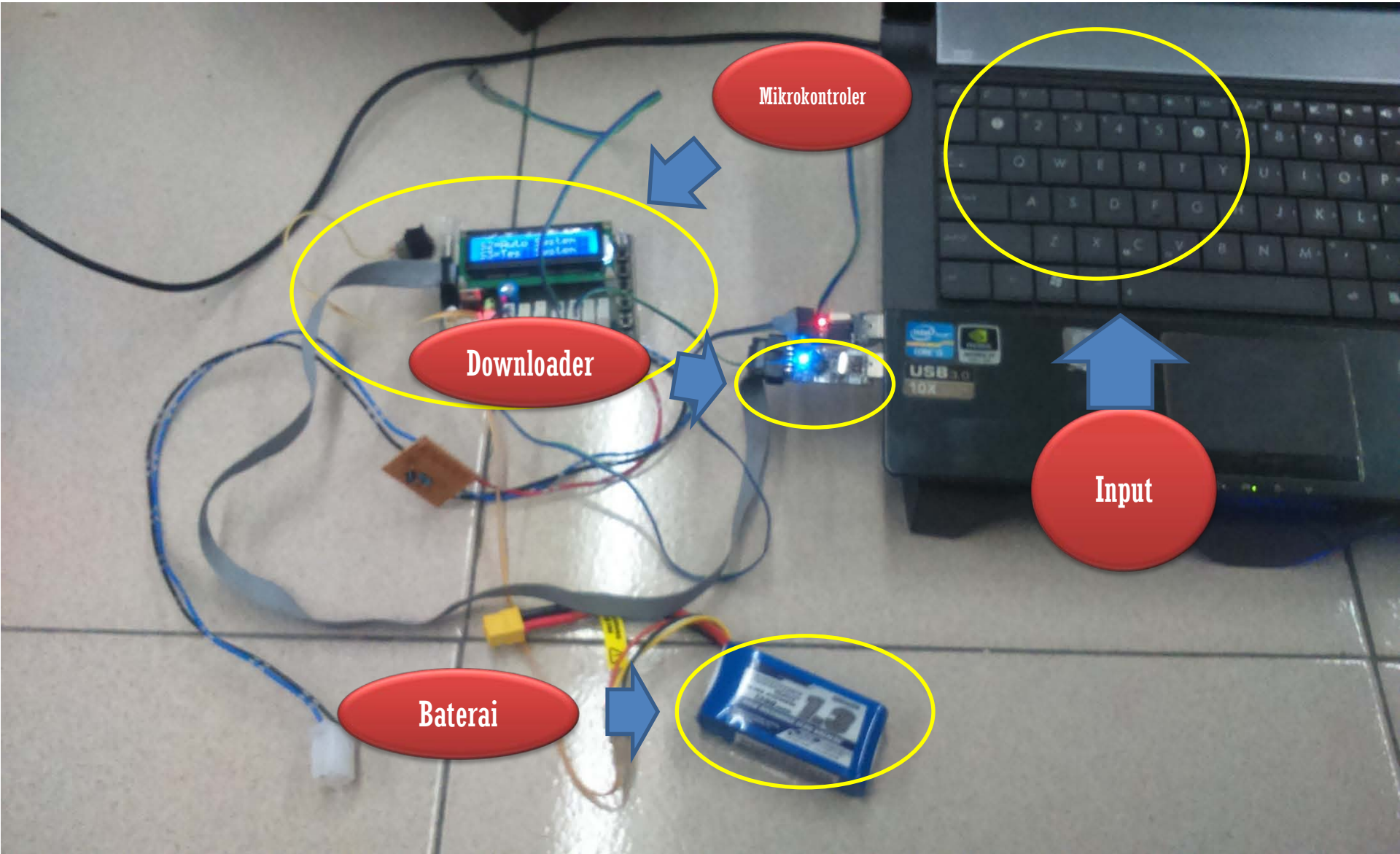


TINJAUAN PUSTAKA





TM



METODE PENELITIAN





Diagram alir

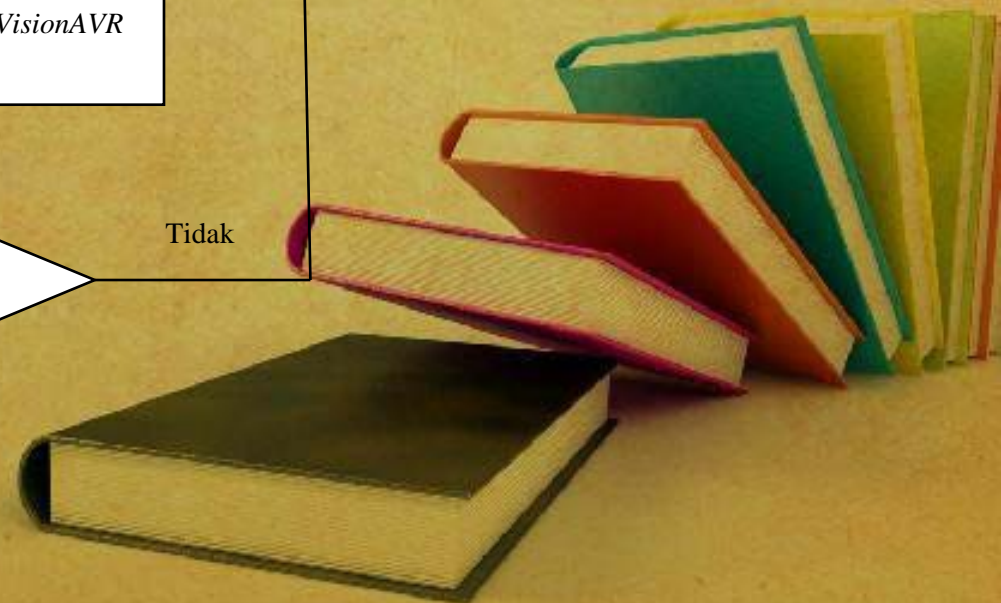
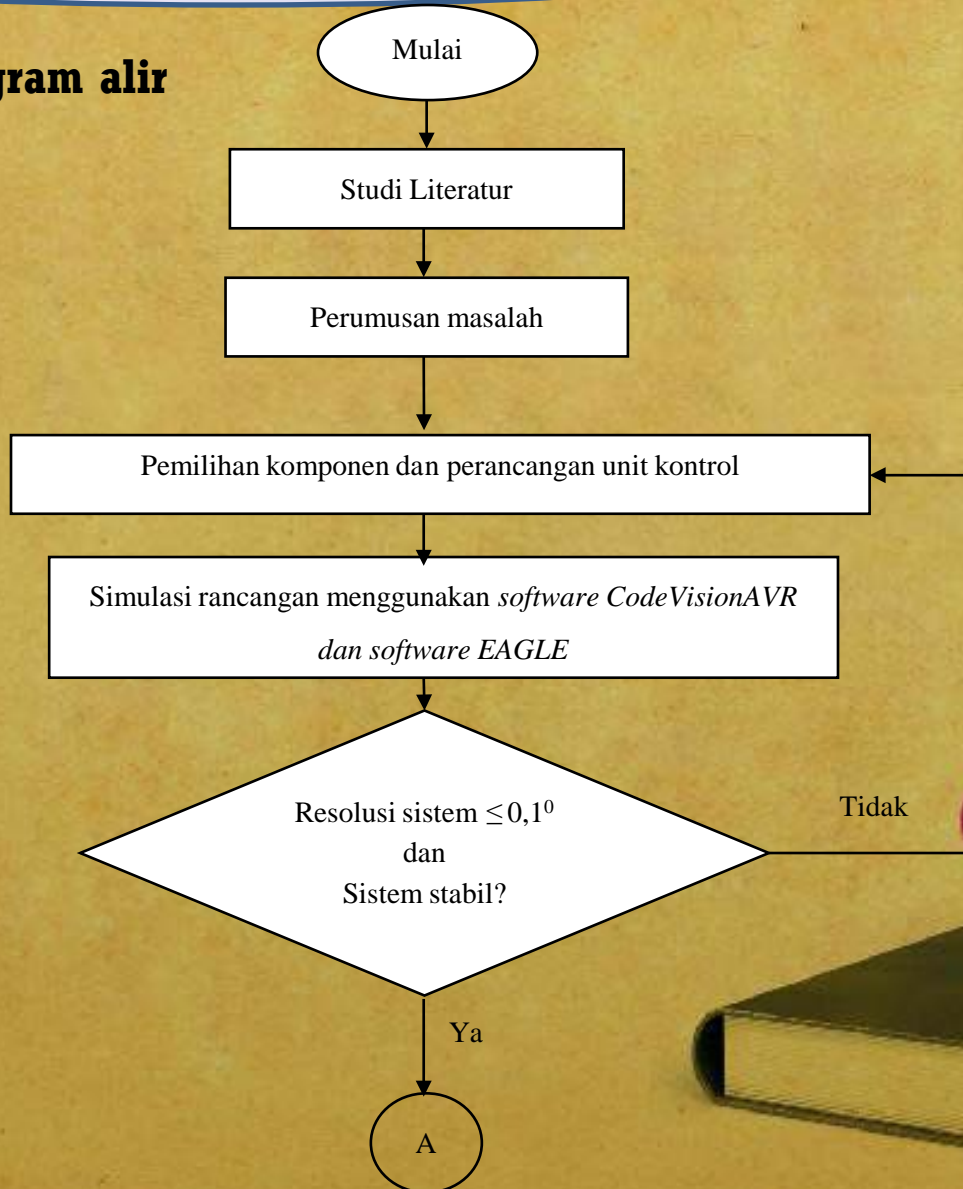
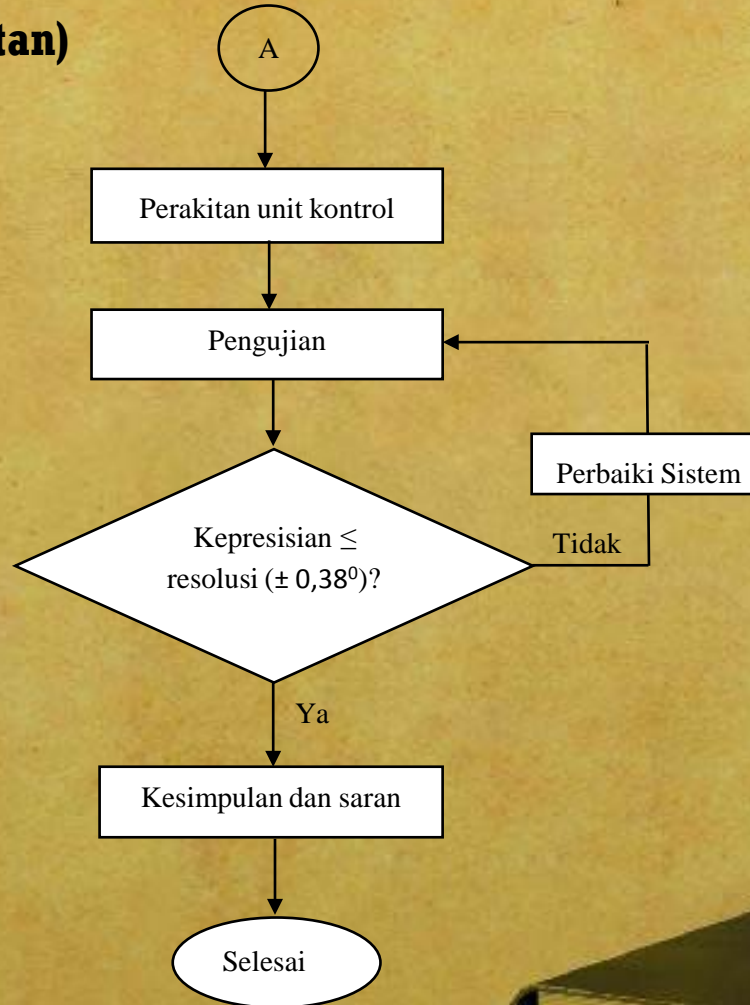




Diagram alir (lanjutan)



PERANCANGAN MINIMUM SISTEM



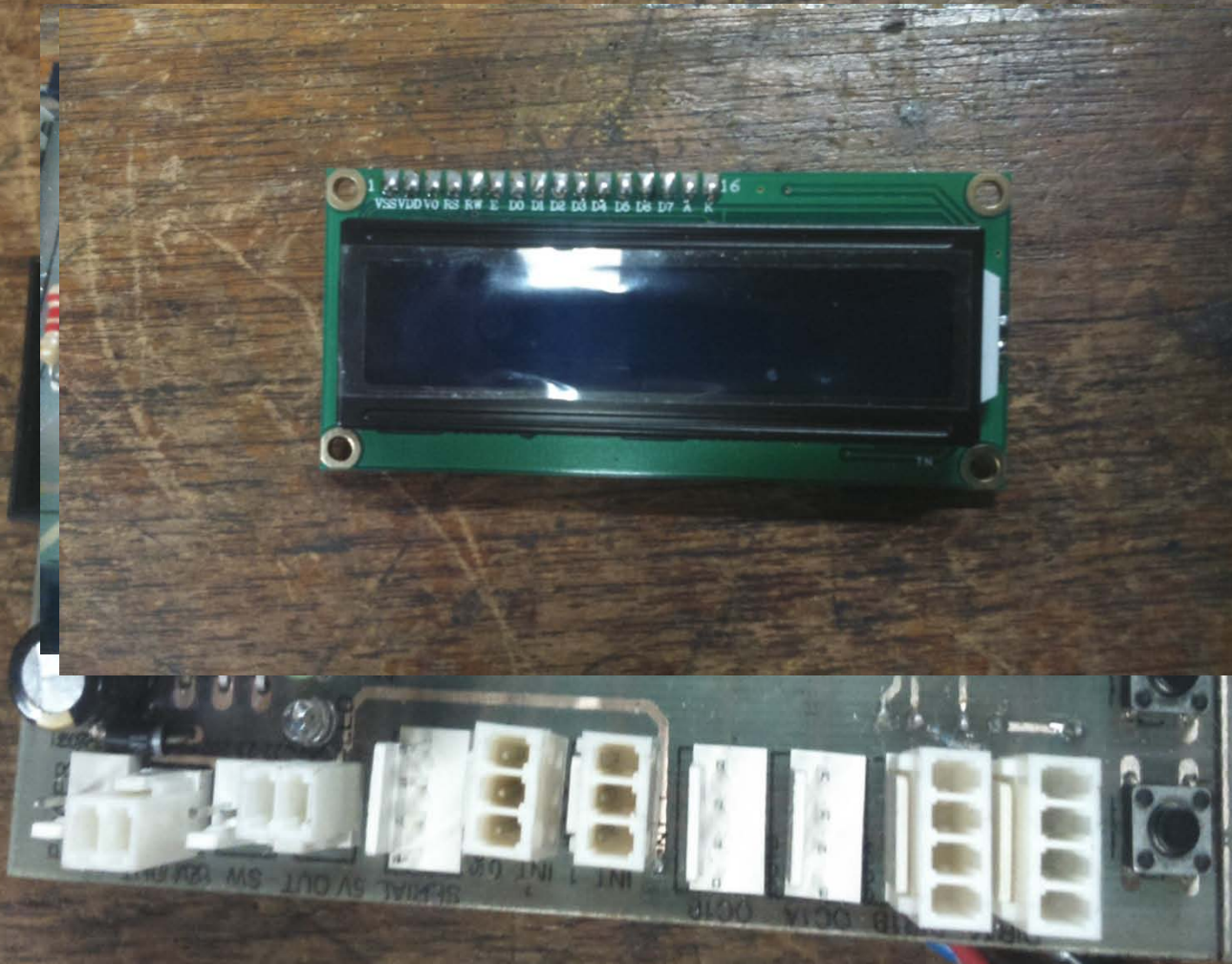
**PERANCANGAN
ENGEMBANGAN
RODUK**
TEKNIK MESIN - INSTITUT TEKNOLOGI SEPULUH NOPEMBER



TM



Perancangan Mikrokontroler





Perancangan Mikrokontroler



Relay





Perancangan *Interface*



Form1

Nilai X :

Send

Nilai Y :

Exit

COM Port :

SET

>> 2 Digit Input Data untuk Nilai X dan Y

Ex : X= +23 dan Y=20
X= -23 dan Y=20

PERAKITAN MINIMUM SISTEM

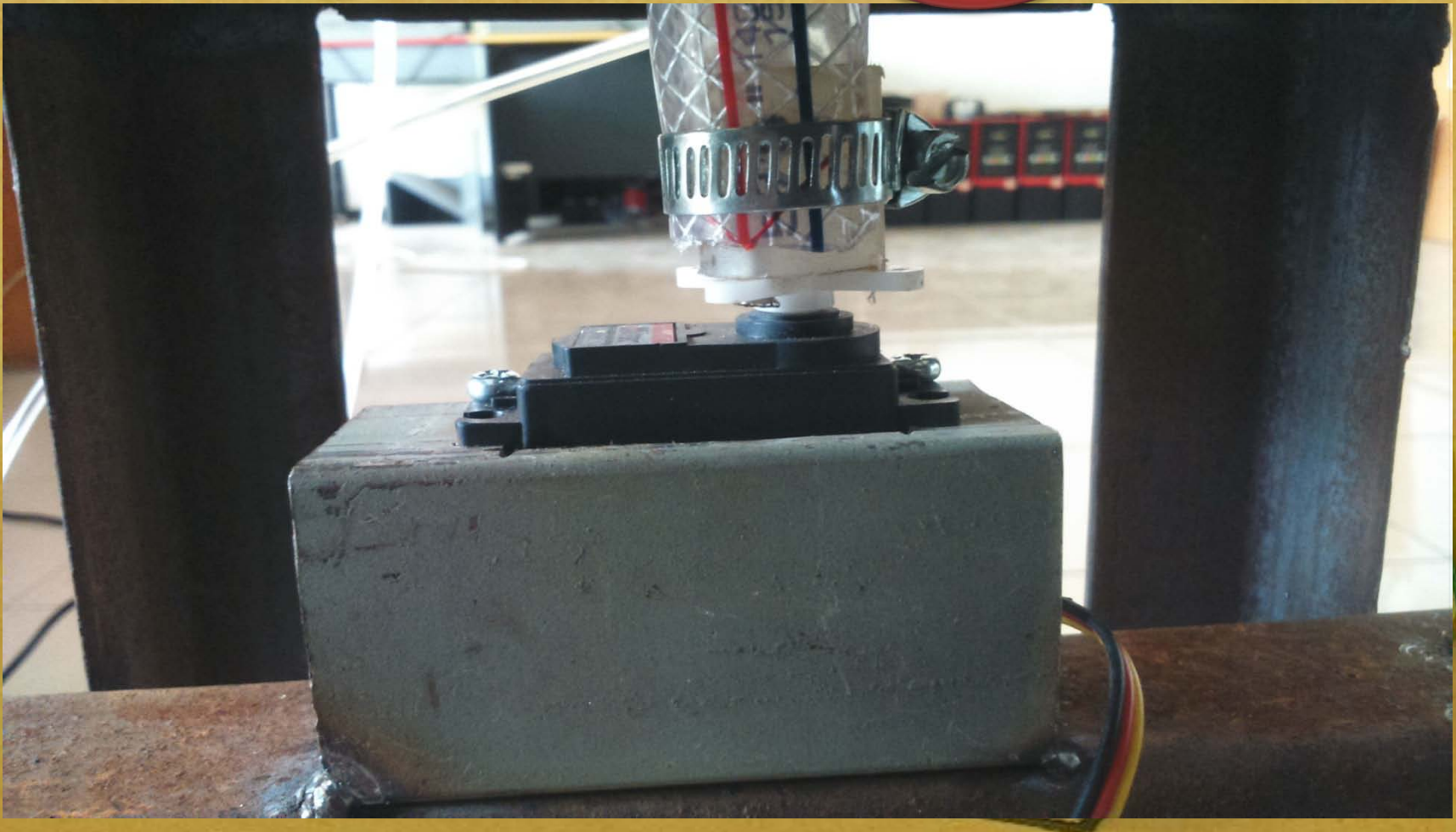




Perakitan Mikrokontroler



Motor Servo
Rotasi

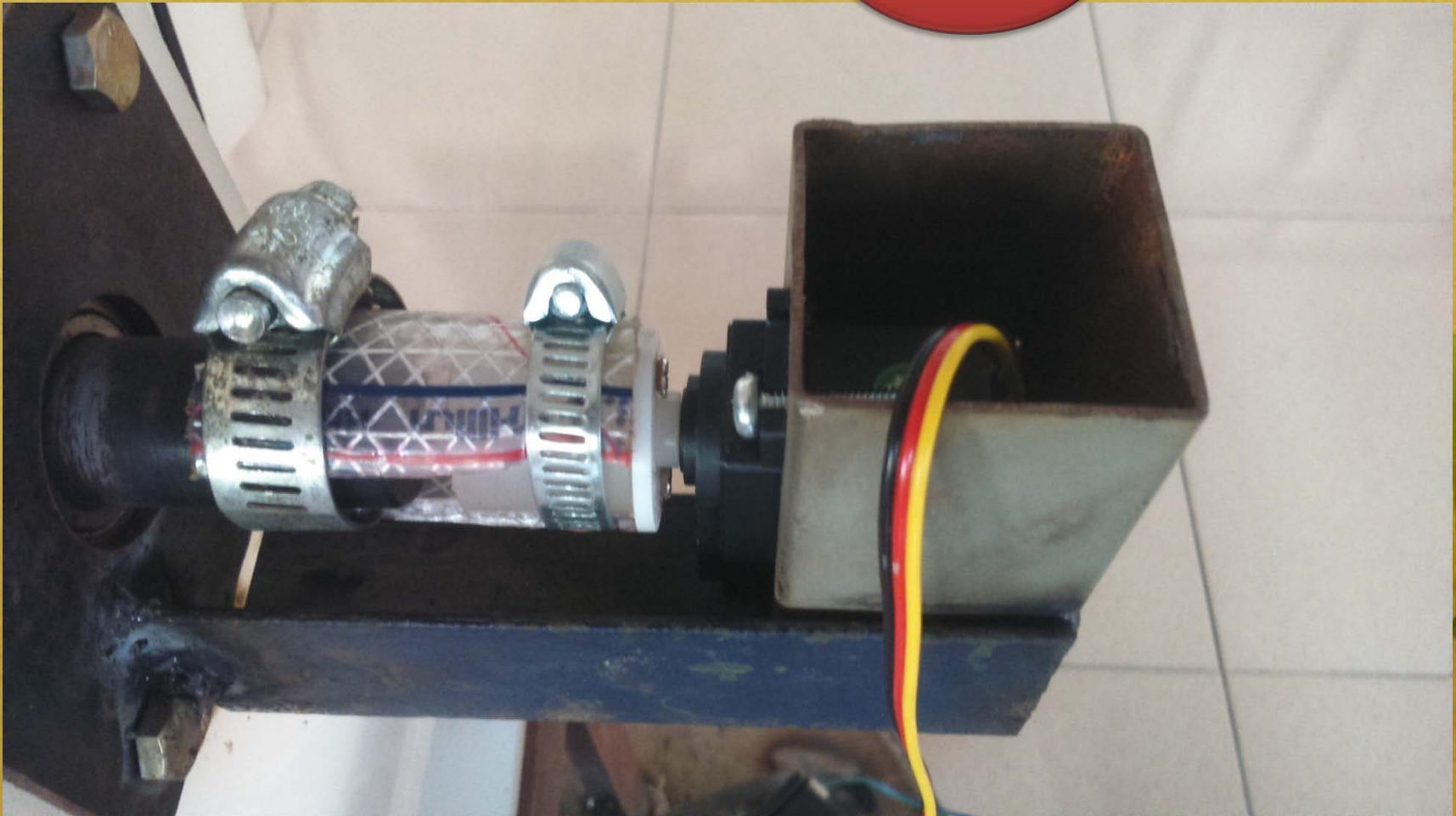




Perakitan Mikrokontroler



Motor Servo
Inklinasi





Perakitan Sistem Kontrol



Perakitan
Sistem
Kontrol



RESOLUSI SISTEM





Kepresisian Sistem Kontrol



ATMega32 memiliki nilai ADC 10 bit atau sebesar 2^{10} satuan data

Tegangan referensi analog adalah tegangan input yang digunakan untuk pengoperasian ADC. Tegangan ini berupa Vcc sebesar 5V.

Dari hasil perhitungan tersebut, *range* tegangan masuk sistem ADC sebesar 0 hingga 4,99511719 V

Tegangan input maksimum dapat diketahui dari perhitungan berikut

$$b_1 2^{-1} + b_2 2^{-2} + \dots + b_n 2^{-n} = \frac{V_{in}}{V_R}$$

$$2^{-1} + 2^{-2} + \dots + 2^{-10} = \frac{V_{in}}{5}$$

$$V_{in} = 5 \times 0,999023438$$
$$V_{in} = 4,99 \text{ Volt}$$





Kepresisian Sistem Kontrol



Nilai ADC maksimal dapat diperoleh dalam bentuk desimal dengan perhitungan sebagai berikut:

$$\begin{aligned} N &= INT \left(\frac{V_{in}}{V_R} 2^n \right) \\ N &= INT \left(\frac{4,99}{5} 2^{10} \right) \\ N &= INT (1023,9) \\ N &= 1023_{10} \end{aligned}$$

ADC memiliki nilai maksimum sebesar 1023 dalam bilangan desimal. Nilai ini diperoleh ketika tegangan yang masuk dalam sistem ADC mencapai nilai maksimum, yaitu sebesar 4,99 volt





Kepresisian Sistem Kontrol



Perhitungan resolusi ADC dapat diketahui dari rumus berikut, dimana tegangan referensi adalah nilai tegangan input ($V_{cc} = 5 \text{ V}$)

$$\Delta V = V_R \times 2^{-n}$$

$$N = INT \left(\frac{V_{in}}{V_R} 2^n \right)$$

Hal tersebut menunjukkan bahwa perubahan tegangan input sebesar 4,883 mV menyebabkan perubahan nilai ADC sebesar 1 bit.

Perubahan nilai ADC yang dihasilkan oleh perubahan tegangan input sebesar 4,883 mV dapat diketahui dengan perumusan sebagai berikut.



Resolusi Potensiometer



Nilai ADC Motor
Servo arah
Inklinasi

$$Y = m.z + c$$

Nilai ADC Motor
Servo arah
Rotasi

$$X = m.z + c$$

Dimana :

x = nilai potensiometer pada arah rotasi

y = nilai potensiometer pada arah inklinasi

m = nilai gradient

c = nilai konstanta

z = sudut



Contoh perhitungan menentukan nilai potensiometer pada motor servo arah rotasi sudut -60° , sudut 0° dan sudut $+60^{\circ}$

Menentukan nilai c pada sudut 0°

$$x = m.z + c$$

$$880 = m(0^{\circ}) + c$$

$$880 = c, \text{ nilai } c \text{ adalah } 880$$

Menentukan nilai m pada sudut $+60^{\circ}$

$$x = m.z + c$$

$$970 = m(+60^{\circ}) + 880$$

$$970 - 880 = m(+60^{\circ})$$

$$90 = m(+60^{\circ})$$

$$m = 1,5$$

sehingga didapatkan :

$$x = 1\frac{1}{2}.z + 880$$





Hubungan antara ADC, Sudut, dan Voltase



Dari kalibrasi nilai potensiometer pada motor servo arah rotasi sudut -60° , sudut 0° dan sudut $+60^\circ$

Nilai potensiometer didapat

- 790 pada sudut -60°
- 880 pada sudut 0°
- 970 pada sudut $+60^\circ$

Untuk mencari sudut:

$$Z = \frac{x - c}{m}$$

- 790 pada sudut -60°

$$\frac{790}{1024} \times 100\% = 77.148\%$$
$$\frac{77.148\%}{100} \times 5V = 3.857V$$

- 880 pada sudut 0°

$$\frac{880}{1024} \times 100\% = 85.937\%$$
$$\frac{85.937\%}{100} \times 5V = 4.296V$$

- 970 pada sudut $+60^\circ$

$$\frac{970}{1024} \times 100\% = 94.726\%$$
$$\frac{94.726\%}{100} \times 5V = 4.736V$$



Contoh perhitungan menentukan nilai potensiometer pada motor servo arah inklinasi sudut 0^0 dan 60^0

Menentukan nilai c pada sudut 0^0

$$y = m.z + c$$

$$900 = m(0^0) + c$$

$$900 = c, \text{ nilai } c \text{ adalah } 900$$

Menentukan nilai m pada sudut 60^0

$$y = m.z + c$$

$$960 = m(+60^0) + 900$$

$$960 - 900 = m(+60^0)$$

$$60 = m(+60^0)$$

$$m = 1$$

sehingga didapatkan :

$$y = 1.z + 990$$





Hubungan antara ADC, Sudut, dan Voltase



Dari kalibrasi nilai potensiometer pada motor servo arah rotasi sudut 0° dan sudut $+60^\circ$

Nilai potensiometer didapat

- 900 pada sudut 0°
- 960 pada sudut $+60^\circ$

Untuk mencari sudut:

$$z = \frac{y - c}{m}$$

- 900 pada sudut 0°

$$\frac{900}{1024} \times 100\% = 87.890\%$$
$$\frac{87.890\%}{100} \times 5V = 4.394V$$

- 960 pada sudut $+60^\circ$

$$\frac{960}{1024} \times 100\% = 93.75\%$$
$$\frac{94.726\%}{100} \times 5V = 4.6875V$$

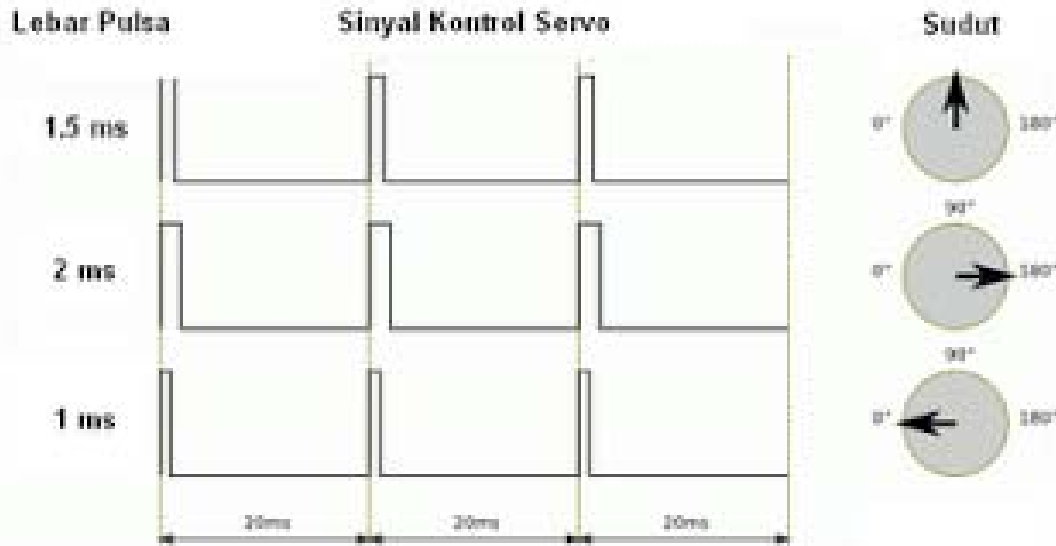




Resolusi Motor Servo



Untuk mendapatkan resolusi motor servo, diperlukan beberapa pengaturan. Pengaturan tersebut dilakukan sedemikian rupa sehingga mempunyai frekuensi 50Hz



OCR 862 untuk
sinyal 1ms, n
mengatur posis
-90 der

Untuk
sinyal 2ms,
atur posisi
derajat.

OCR 1978 untuk
menghasilkan sinyal 1.5ms,
maka Servo mengatur posisi
untuk sudut 0 derajat.



Diagram alir

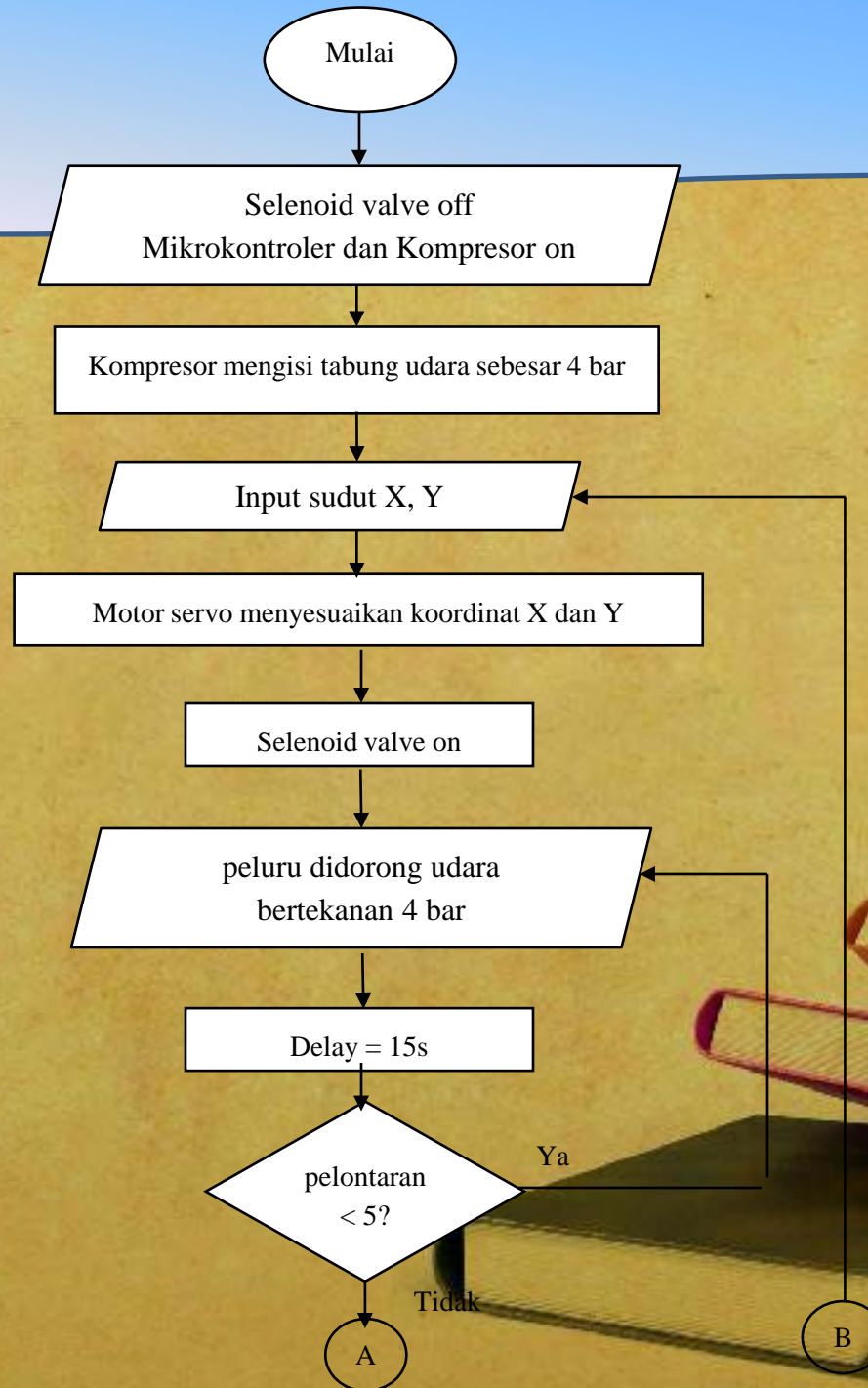
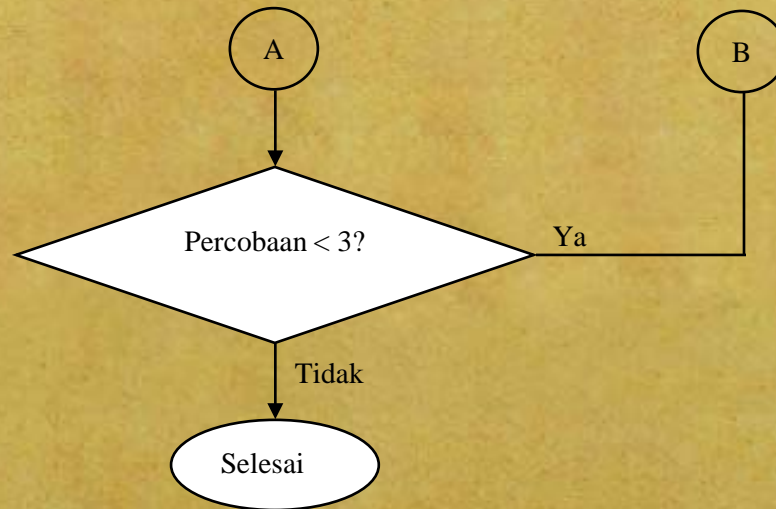




Diagram alir

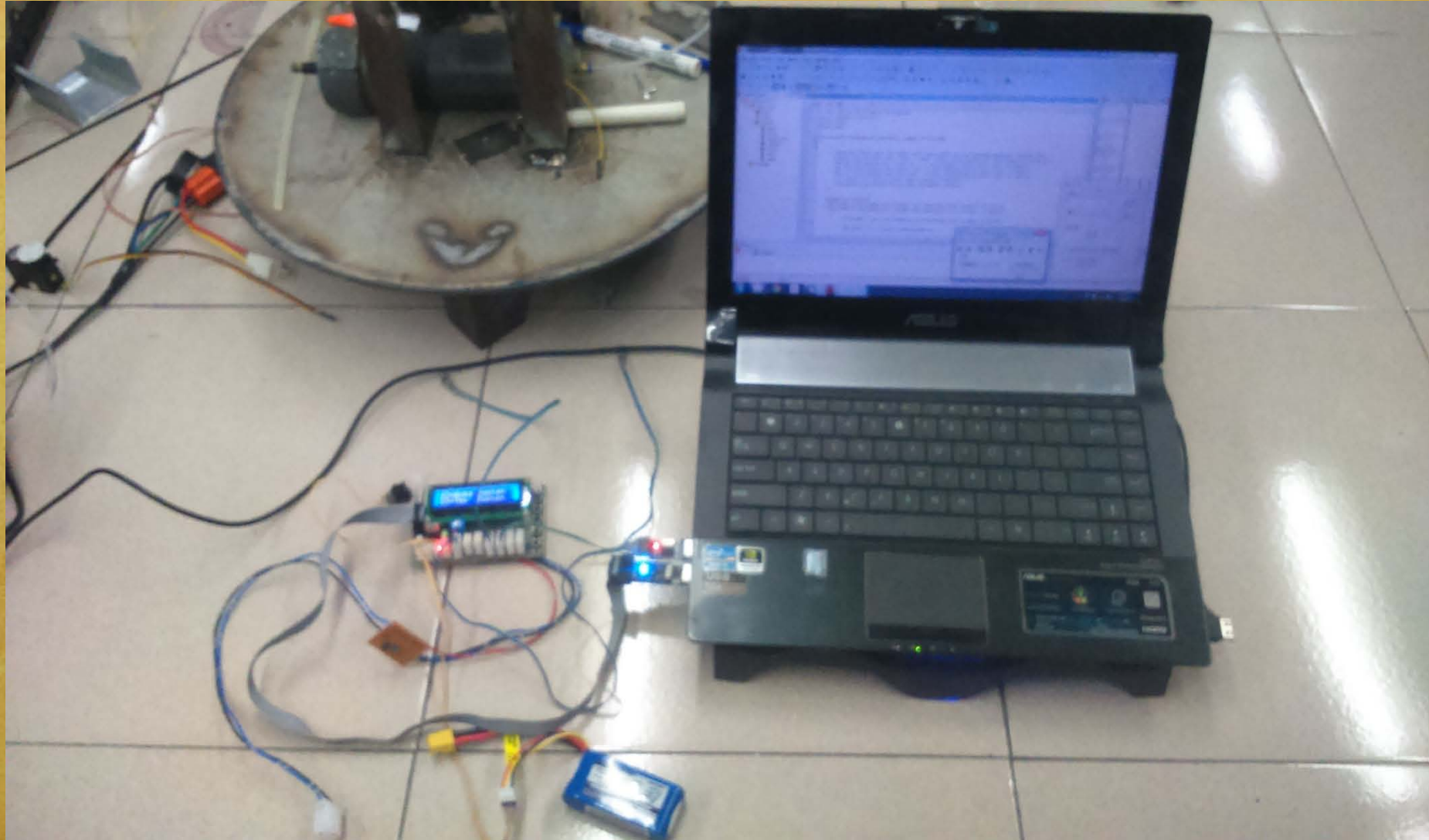


PENGUJIAN





PENGUJIAN





PENGUJIAN



S2 Auto Sistem



S3 Tes Sistem





PENGUJIAN



S2 Auto Sistem

Form1

Nilai X :

Send

Nilai Y :

Exit

COM Port :

SET

>> 2 Digit Input Data untuk Nilai X dan Y

Ex : X= +23 dan Y=20
X= -23 dan Y=20

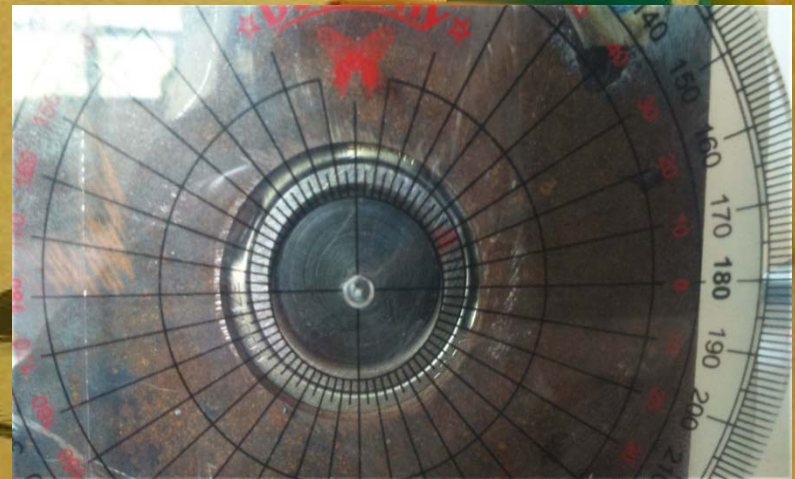




PENGUJIAN pertama



Pengujian titik
(-60° , 30°)





PENGUJIAN kedua



Pengujian titik
(0° , 30°)

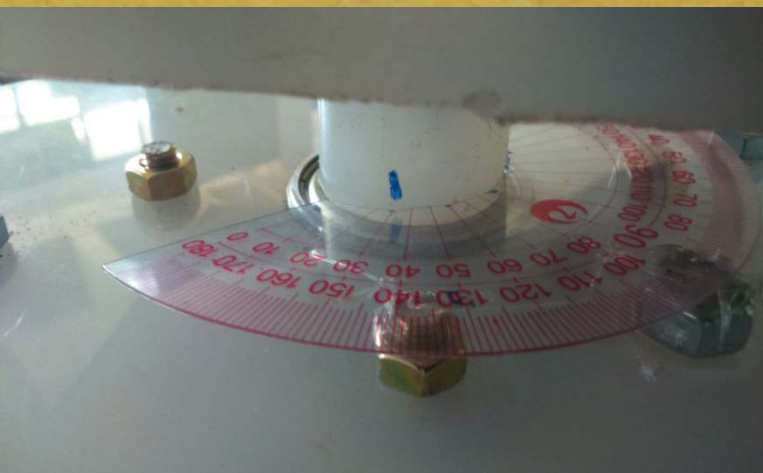




PENGUJIAN ketiga



Pengujian titik
($+60^{\circ}$, 30°)



KESIMPULAN DAN SARAN



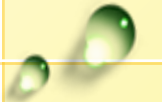


Kesimpulan

- Masih terjadi banyak *noise* dalam percobaan tes sudut menggunakan potensiometer, sehingga nilai yang didapat berubah-ubah.
- Pengambilan nilai pembacaan tidak sesuai dengan yang diharapkan, dikarenakan perubahan alat pembacaan menjadi manual.
- Dari ketiga pengujian tersebut dapat dilihat pembacaan sudut dengan bantuan busur derajat sudah akurat, tetapi kurang cermat.
- Kecermatan pembacaan dengan busur derajat masih kurang dibandingkan dengan pembacaan menggunakan potensiometer pada motor servo.

Saran

- Masih terjadi banyak *noise* dalam percobaan tes sudut, sehingga nilai yang didapat berubah-ubah, oleh karena itu sebaiknya proses instalasi dilakukan secara hati-hati.
- Pembuatan alat yang *otomatis*, sehingga alat tidak perlu melakukan proses pengimputan data melalui komputer.
- Perlu penataan kembali letak komponen-komponen perangkat kerasnya.



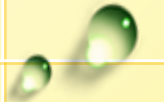
Terima kasih

Fine alla fine



TM





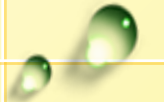
Terima kasih

Fine alla fine



TM





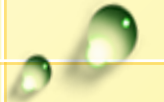
Terima kasih

Fine alla fine



TM





Terima kasih

Fine alla fine



TM

